MASTER EN SCIENCES INFORMATIQUES

Implementation of exchange rate model using text mining techniques

Frogneux, Etienne; Ronvaux, Gilles

Award date:
2006

Awarding institution:
Universite de Namur

Facultés Universitaires Notre-Dame de la Paix, Namur
Institut d'Informatique
Année académique 2005-2006

Implementation of exchange rate model using text mining techniques

Etienne Frogneux - Gilles Ronvaux

Mémoire présenté en vue de l'obtention du grade de Maître en Informatique

Résumé

Ce document est divisé en deux parties. Tout d'abord, l'état de l'Art qui donne un aperçu du procédé de data mining et de l'utilité de celui-ci. Différentes techniques utilisées en data mining sont présentées. Ensuite, nous présentons un cadre d'analyse pour le text mining (extension du data mining). Les différente étapes du processus de text mining sont ensuite décrites et illustrées par des exemples. La deuxième partie de ce travail concerne le projet qui a été réalisé en collaboration avec l'UTS (University of Technology of Sydney), en Australie. L'objectif de ce projet est d'utiliser des nouvelles économiques provenant de l'internet pour modéliser le taux d'échange entre deux monnaies (dans notre travail, nous considérons uniquement l'Euro et le Dollar américain). Les différentes étapes du projet sont décrites. Nous soulignons aussi quelques problèmes rencontrés lors de l'élaboration du projet ainsi que les solutions que nous avons proposées pour résoudre ces problèmes. Pour terminer, nous présentons les résultats de notre projet et une conclusion faisant suite à ces résultats.

Mots Clés: data mining, text mining, taux d'échange, KDD, Dollar américain, Euro

Abstract

This document is divided in two parts. First, the Sate of The Art gives a view of the data mining process and the utility of it. Different techniques used in data mining are introduced. Following this, a framework of text mining, an extension of data mining, is presented. The different steps of the text mining process are described using examples. The second part concerns the project that has been realized in collaboration with the Faculty of Information Technology, University of Technology, Sydney Australia. The goal of this project is to use news articles and economic data from the internet to model the exchange rate changes between currencies (in our study, only between the Euro and the US Dollar). The different steps of the project are described. We also point problems we have encountered and solutions we propose in order to solve them. Finally, we present the results of our project and the conclusion following those results.

Keywords: data mining, text mining, exchange rate, KDD, US Dollar, Euro

Acknowledgements

Nous tenons à remercier toutes les personnes qui, d'une manière ou d'une autre, ont permis la réalisation de ce travail.

Nous pensons spécialement à Madame le Professeur Monique Noirhomme, de l'université de Namur, qui a accepté d'être la promotrice de notre mémoire et qui, par sa connaissance du sujet, ses remarques constructives et ses conseils, nous a permis de réaliser ce travail.

Merci également à Monsieur Etienne Cuvelier pour son aide en ce qui concerne la partie sur les Support Vector Machines et à Vincent Frogneux pour avoir relu et corrigé notre travail.

We would like to acknowledge the Associate Professor Simeon Simoff for his supervision during our traineeship at the University of Technology, Sydney. We also thank him for the documentation and the advices he gave us.

We would also like to acknowledge the doctor Debbie Zhang for her daily help, her kindliness and her numerous advices.

Nous voudrions finalement remercier nos familles, nos amis pour leur soutien et leur aide durant la réalisation de ce mémoire et tout au long de nos études.

Contents

List of Figures

List of Tables

Glossary

ANN Artificial Neural Network, 46, 50

Coreferences Multiple descriptions of the same event, 32

Gross Domestic Product (GDP) total goods and services produced by a nation during one year excluding payments on foreign investments (Economics) ., 78

Gross National Product (GNP) total value of goods and services produced by a country in a given period (generally one year)., 78

Independent Identically Distributed said of a collection of random variables that all have the same distribution and are independent of each other. This is often abreviated as *iid.*, 54

Inflation increase in the supply of money in relation to the amount of goods available resulting in a rise in prices., 78

Interest Rate rate of the money that is paid for a financial service (ex:loan), 78

MCP Neuron McCulloch and Pitts' Neuron, 49

SRM Structural Risk Minimization, 57

Template document used as a foundation for new documents having a uniform style., 31

Unemployment Rate percent of unemployed persons from within the general population., 78

VC Theory Vapnik-Chervonenkis theory, 57

Introduction

The variability of the exchange rate has always been and will always be a source of incertitude for all economists. It would be therefore valuable to know the value of the exchange rate between two currencies (for example, between the euro and the US dollar) with two days of advance. Economists have tried and still try to find some theories to satisfy this desire. Despite several years of research, they do not reach to predict the future value of the exchange rate. Nonetheless, some theories have demonstrated that news which are released on the Internet have an impact in short term on the exchange rate between two currencies, especially between the Euro and the US dollar.

At the University of Technology in Sydney, they are working on this theme and they are trying to find an algorithm to help economists, and therefore to predict on short term the value of the exchange rate. Once we started working on the project, we wondered if it was possible to use **an algorithm which is able to predict on short term the value of the exchange rate between the Euro and the US dollar by using text-mining techniques and by analyzing news from the Internet.** The application that we had to continue includes two main parts. The first one is the automatic classification of the data news by using Text-Mining techniques (is the selected news positive for the US dollar currency or not ?). The second one is the exchange rate prediction by using the results obtained by the first part. Personally, we have essentially worked on the first part, namely, the classification with Text-Mining techniques.

Our work will be divided into two parts. The first part is devoted to the state of the Art (the first three chapters). In this part, a first chapter exposes the Knowledge Discovery in Databases and Data Mining where some techniques will be explained. In the second chapter, we present the Text-Mining process, the in-

formation extraction and the information mining. The third chapter is devoted to the Support Vector Machine prediction.

In the second part of our work, we explain what we have done during our traineeship at the University of Technology in Sydney. We present the project in which we took part (chapter 4). After the description of the project, we present the different functionalities and interfaces we have developed (chapter 5). The chapter 6 is dedicated to the results we obtained with the prediction algorithm. The last chapter shows concretely the different points of our contribution to the project.

To conclude this work, we introduce the concepts that could be interesting to consider in order to improve the results.

Chapter 1

Knowledge Discovery in Databases and Data Mining

The amount of data being collected in databases today far exceeds our capability to reduce and analyze data without the use of automated analysis techniques. Many scientific and transactional business databases are growing at a phenomenal rate. Thus, there are too much data, and it will be useful to develop an automatic method to mine data in order to interpret easily results and patterns obtained by this method. This method is called "Knowledge Discovery in Databases".

Sometimes, the words Data Mining and Knowledge Discovery in Databases are used jointly as synonyms, but these words are completely different. In fact, Data mining is only a step in knowledge Discovery in Databases. In this chapter, we are going to define separately the Knowledge Discovery in Databases and Data Mining. Firstly, Knowledge Discovery in Databases will be explained, then we will expound the Data Mining.

1.1 Knowledge Discovery in Databases

Knowledge Discovery in Databases can be defined as *the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [FPSS96]. In other words, the goal of Knowledge Discovery in Databases is to try to "learn" from data some knowledge expressions which explain that data. Essentially, the aim is to generate rules, patterns, models from the data.*

It will be interesting to explain some terms of this definition :

- **Data**: Data is a set of facts, called F. It is also interesting to remark that large amounts of data are required to provide significant information and to derive additional useful knowledge.

- **Pattern**: A pattern is an expression E represented by a language L describing facts in a set F_E. This set is a subset of F.

$$F_E \subset F$$

- **Process**: A Knowledge Discovery in Databases process is a multi-step process. That implies some steps, namely data preparation, search for patterns, knowledge evaluation, and refinement involving iteration after modification. This process is qualified "non-trivial" to have some degree of search autonomy.

- **Validity**: The patterns discovered in the data should be again valid with new data with some degree of certainty. The function $c = C(E,F)$ is a measure of certainty and this function maps expression in the language L to a partial or total space called M_C. This certainty measure can be allotted to the expression E described by the language L about a subset $F_E \subset F$.

- **Novel**: The patterns must be novel. A function N(E,F) measures the changes in data or knowledge and thus determine if the patterns are novel or not. The result of this function is a measure of degree of novelty, but can be a boolean to state if the patterns are novel or not.

- **Potentially Useful**: Sometimes, useful actions can be brought to patterns. To measure these useful actions, an usefulness function exists. This function maps expression E (in the language L) to a partial or total space M_U. In brief, $\mu = U$(E,F).

- **Ultimately Understandable**: For humans, it is preferable to have understandable patterns. Thus, for this reason, Knowledge Discovery in Databases will provide understandable patterns to humans with the intention to facilitate the understanding of the data. Once again, a function exists to evaluate the "simplicity measure". This function S maps expressions E (enunciated by the language L) to a partial or total space M_S. In brief, s = S(E,F).

1.1.1 Characteristics of the Knowledge Discovery in Databases

The results of the knowledge Discovery in Databases are called "discovered knowledge". This discovered knowledge can be used to make predictions or classifications about new data, explain existing data, summarize the contents of a large database, to facilitate decision making or to visualize logical data to aid humans in discovering deeper patterns.

Discovered knowledge can be characterized by its form, its representation and its degree of certainty.

As far as the form of a discovered knowledge, we can divide it into two categories:

- **quantitative discovery** that requires mathematical equations using numeric field values.
- **qualitative discovery** that tries to find a logical relationship among fields.

Frequently, simple rules are used to enunciate qualitative and quantitative discoveries. These rules can be "$X > Y$" or "A *implies* B". However, these discoveries can also be described by other more complex forms.

About the representation of the discovered knowledge, it exists several representation forms in accordance with the intended user. We can distinguish two user categories, namely the humans and computers. For human end user, the used representations are the natural language, formal logics, and visual depictions of information. Contrariwise, for computer, we must privilege representations like programming languages and declarative formalisms.

Finally, the last characteristic of the discovered discovery concerns the uncertainty of the patterns discovered by the Knowledge Discovery in Databases process. In fact, in the discovered patterns, a lot of noise, missing fields,... are present. It is important to pay attention that the degree of certainty of the information that we discover from the data is not always excellent. Thus, we must analyze information cautiously.

1.1.2 The Knowledge Discovery in Databases process

As explained above, Knowledge Discovery in Databases is defining as a process. The exact definition of the Knowledge Discovery in Databases process is: "*Process of using data mining methods (algorithms) to extract (identify) what is deemed knowledge according to the specifications of measures and thresholds, using the database F along with any required preprocessing, subsampling, and transformations of F*" [FPSS96].

This Knowledge Discovery in Databases process can be defined as iterative and interactive. This process is iterative because the Knowledge Discovery in Databases process can involve significant iterations (see figure 1.1) and can contain loops between any two steps. On the other hand, the process is interactive because numerous decisions can be made by the user. [Figure 1.1] shows an overview of the steps that compose the Knowledge Discovery in Databases process.

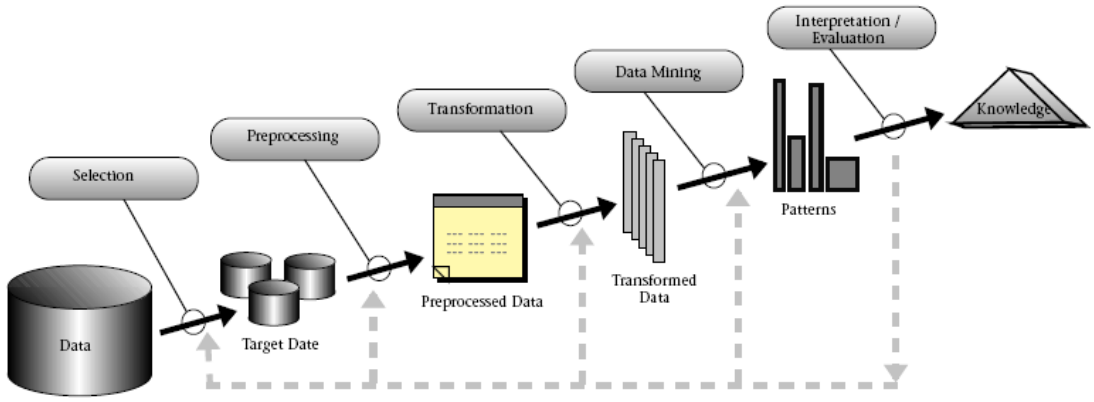

Figure 1.1: Knowledge Discovery in Database process [FPSS96]

According to [FPSS96], this process includes nine steps:

1. The first step is the step related to the understanding of the application domain in which the process will be executed. Moreover, the identification (from the user's point of view) of the goal of the Knowledge Discovery in Databases process is realized during this step.

2. The second step is creating a target data set. In other words, from a database (for example), a data set is selected. We can also take a subset of variables or a data sample. It is on this set that the process will be performed.

3. The third step is called "data cleaning and preprocessing". Some operations will be performed on data set built by the previous step. These operations are necessary because data in the real world is dirty. Therefore, operations are required because no quality in data implies no quality mining results. These workings are mainly designed to eliminate noise and errors in the data. Furthermore, this step can be also used to fill fields whose the value is missing, to smooth noisy data, to resolve inconsistencies,...

4. The fourth step is named "data reduction and projection". The aim of this step is to obtain a reduced representation of the data set. Generally, this data set is too large but data reduction must produce the same results as the primary data.

5. The fifth step is the selection (by the user) of the data mining method that will be used in this process. These methods can be classification, clustering, regression, ... This method is chosen according to the goals of the Knowledge Discovery in Databases.

6. The sixth step is called "exploratory analysis and model and hypothesis selection". The purposes of this step consist to the choice of the data mining algorithm(s) that will be executed and the selection of the method(s) to be used to search for data patterns. The models and the values of the parameters will be decided during this step.

7. The seventh step is the specific step of "Data Mining". During this step, the aim is the searching for patterns in the data set. We will deepen the data mining presentation in another part of this work.

8. The eighth step is "interpreting mined patterns". Once patterns are discovered, it must interpret results. Nevertheless, it is possible to return to any of steps 1 through 7 for another iteration of the process. Visualization can be used to interpret results because sometimes, it is easier to interpret a graph than numerous numbers.

9. The last step is "acting on the discovered knowledge". During this step, discovered knowledge could be associate to another knowledge. This knowledge can be knowledge from another system.

1.1.3 Characteristics of the discovered patterns

It would be interesting to analyze patterns that the Knowledge Discovery in Databases process produces. Knowledge Discovery in databases owns four main characteristics [FPSM91]:

1. ***High level Language***
   The first characteristic is that discovered knowledge is represented in a high-level language such as:

   $$\begin{gathered} \textit{If } \text{Age} < 25 \text{ and Driver-Education-Course} = \text{No} \\ \textit{Then } \text{At-fault-accident} = \text{Yes} \\ \textit{With } \text{likelihood: } 0.2 \text{ to } 0.3 \end{gathered}$$

   From this example, we can observe that patterns can be understood easily and used like that by people. Moreover, another program can used these patterns as input.

2. ***Accuracy***
   The second characteristic is "Accuracy". The degree of certainty is essential for system or user. In fact, the granted trust to the discovery depends on the certainty degree of patterns.

   Accuracy requires some factors:

   - integrity of the data
   - size of the sample
   - degree of support from available domain knowledge

   It is interesting to remark that patterns would become unjustified if accuracy was not sufficient enough. Thus, with a insufficient accuracy, knowledge will not be reliable.

3. ***Interesting results***

   The third characteristic of the Knowledge Discovery in Databases process is named "interesting results". A big number of patterns are extracted from the process, but all of them are not interesting. Therefore, patterns are considered as interesting if they are novel, potentially useful and that the discovery process is nontrivial. Using the example presented above:

   $$\textit{If}\ \text{At-fault-accident} = \text{yes}\ \textit{Then}\ \text{Age} > 16.$$

   From this finding knowledge, this discovery can be qualified as previously unknown and potentially useful for the system. But, for a user, this discovered pattern would be uninteresting. Thus, this information would not be considered as a discovery. In conclusion, novelty and usefulness without a nontrivial process are not enough to have a pattern qualified of discovered knowledge.

4. ***Efficiency***
   The last characteristic is "Efficiency". The discovery process can be qualified as efficient since this process can be implemented by an algorithm and by definition, an algorithm is efficient if the run time is a polynomial function. The efficiency of the discovery process is justified because the run time for a algorithm on a large-sized database is predictable and acceptable.

1.1.4 Database problems

We have seen in the third step of the Knowledge Discovery in Databases process that it is necessary to make some operations on the databases. These operations are necessary because there are some problems in the databases and in order to have a good knowledge, it is necessary to resolve these issues. There exists five main problems:

1. ***Dynamic data***
   The first problem indicates us that the content of a database can be changing. Between two dates, the value of a field can change. In fact, sometimes data are variable, namely for some data, time affects the value. It exists three data categories. The first category is the data that are constant over time.

The Identification number of the National Register belongs to this category. The second class is the data that vary more or less progressively over time. For example, this category includes the weight, the height or the age. At last, the last category includes data that are very inconstant. The value of data changes frequently. The pulse rate belongs to this last category.

2. ***Irrelevant fields***
   Another problem speaks about the relevance of data. Let us take an example. In a hospital, there is a database with all information on the patients. For a doctor, in these information, some data are relevant, but other data are irrelevant. For example, for a doctor, medical data (pulse rate, temperature,...) are relevant. Contrariwise, nonmedical data (postal code, address,...) are irrelevant. Once again, for a doctor, errors in relevant data are more important than in the irrelevant data. In this case, that is not important. In the other side, errors in the relevant data can be bothersome and information that will be discovered can be erroneous.

3. ***Missing values***
   The discovery can be influenced by the absence of some relevant data fields. In fact, in medicine for example, the absence of some patients measurements can affect the doctor diagnostic. To avoid it, the missing attribute will be assigned by a neutral value.

4. ***Noise and Uncertainty***
   The fourth database problem is "Noise and Uncertainty". First of all, the severity of errors is linked with the data type. The possible type of data are real numbers, integer numbers, strings, an element of nominal values. Specially with numeric data types, precision is very important and a main factor for the discovery. Moreover, more noise are present in the data, less the discovery is reliable.

5. ***Missing fields***
   Finally, the last problem concerns the missing fields. It must pay attention when we analyze a database. A database can appear correct even when there are some errors. An example to illustrate this problem: A system is implemented to learn to diagnose malaria from a patient database, but this database does not contain the red blood cell count. For the system, patient is healthy but is it the same in the reality?

1.2 Data Mining

After having introduced the Knowledge Discovery in Databases process, we now focus on the data mining components. The term data mining can be defined as: "*a step in the Knowledge Discovery in Databases process consisting of particular data mining algorithms that, under some acceptable computational efficiency limitations, produces a particular enumeration of patterns E_j over F*" [FPSS96].

Moreover, seeing that the Knowledge Discovery in Databases process is qualified as iterative, some data mining methods can be apply several times in the same process. That is why Data Mining is the core of the Knowledge Discovery in Databases process.

Patterns are what Knowledge Discovery produces. Nevertheless, it is interesting to distinguish a model from a pattern. A pattern can be thought of as instantiation of a model:

$$f(x) = 3x^2 + x \text{ is a pattern}$$
$$f(x) = \alpha x^2 + \beta x \text{ is a model}$$

Thus, data mining involves fitting models to observe data or determine patterns from observed data. Two primary mathematical formalisms can be used in model fitting:

- Statistical approach
  One of the characteristics of the statistical approach is the possibility of non-deterministic effects in the model. For example, in the statistical approach, we can have this function: $f(x) = \alpha x + e$ (where e could be a Gaussian variable).
- Logical model
  Contrary to the statistical approach, the logical model is completely deterministic. $f(x) = \alpha x$ is a deterministic function.

1.2.1 The primary tasks of Data Mining

If we observe Data Mining and knowledge discovery, we can differentiate two types of goals. That is the use of the system that separates these aims:

1. Verification
   The verification task tries to verify the user's hypothesis.

2. Discovery
   As regards discovery, the goal of the system is to discover new patterns. The discovery goal can be divided into two other goals, namely:

   - Prediction
     With some variables or fields that are present in the database, prediction is able to find unknown or future values for variables of interest.
   - Description
     Contrary to prediction, description is centered on the patterns produced by an algorithm. From these results, description finds some human-interpretable patterns. As we can see on the figure 1.2, each goal has these Data Mining methods. Indeed, classification, regression, times series analysis and prediction methods belong to prediction goal. Summarization, clustering, association rules, sequence discovery methods belong to description goal.

Now, we are going to explain quickly these Data Mining methods.

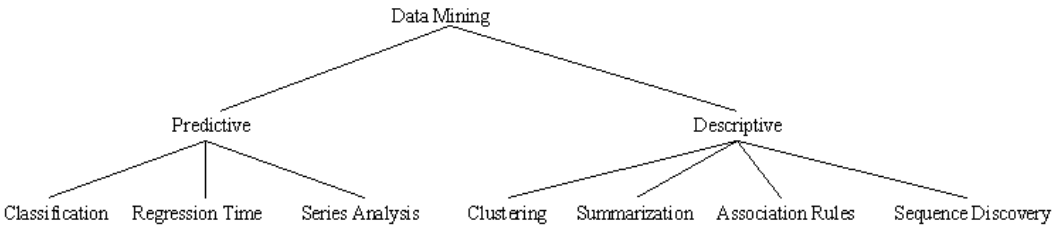

Figure 1.2: Goals of discovery in Data Mining

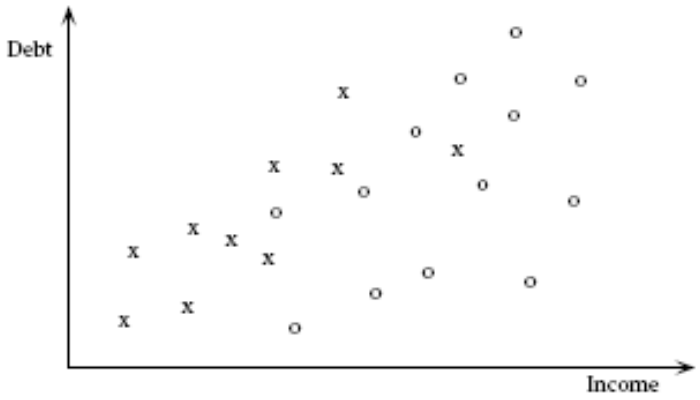

Figure 1.3: Example to illustrate Data Mining Methods. [FPSS96]

1.2.2 Data Mining methods

First of all, for all methods, we are going to illustrate them by using an example. Figure 1.3 shows a simple two-dimensional data. In this graph, each point represents a person who has been a loan in a bank. The horizontal axis represents the income of the person, and the vertical axis represents the total debt of the person. These data have been classified into two classes. The class represented by "x" is persons who have defaulted on their loans, and the symbols "o" represent persons whose loans are in good status with the bank.

Before building some predictive models, it is preferable to understand data, and for that, we use methods with a description goal, namely summarization, clustering, association rules and sequence discovery.

Descriptive methods

Summarization is a descriptive method whose the goal is to find a compact description of a subset of data. Thus, summarization can be considered as an abstraction or generalization of the data. The result of the summarization gives us a general overview of the data as a whole. Usually, summarization uses aggregation of information. At last, let us signalize that it exists different levels of abstraction.

Another descriptive method is ***clustering***. Clustering can be defined as *"a process of partitioning a set of data in a set of meaningful sub-classes, called clusters"* [Zaï99]. [Zaï99] defines also a cluster as *"a collection of data objects that are "similar" to one another and thus can be treated collectively as one group"*. About characteristics of a cluster, it is required to maximize intraclass similarities and to minimize interclass similarities. Once clusters are discovered, a label (corresponding to the name of the cluster) is applied to objects. All objects in a cluster are summarized to form the description of this class. On the figure 1.4, we can see a clustering applied to the example of loan.

Clustering can be classified according to two manners:

1. Supervised classification, also called classification is used when we know the class and the number of classes.

2. Unsupervised classification (called clustering) is employed when the classes are not known and the number of classes is not determined in advance.

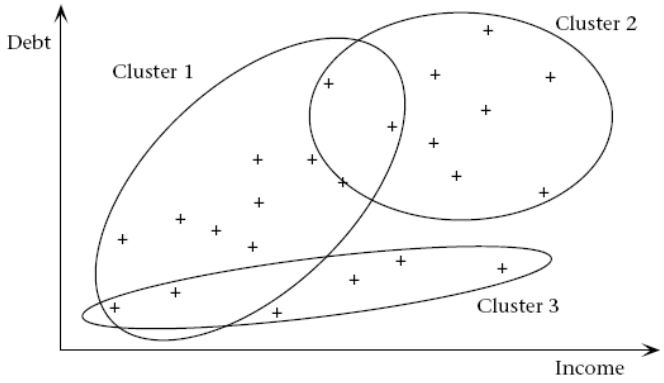

Figure 1.4: Example of clustering. [FPSS96]

The third descriptive method is ***Association rules***. The purpose of this method is to understand association analysis. An association rule reveals the associative relationships among objects. Association rule tries to find association, correlation,... among sets of items in databases. To be formal, we need to have some definitions before to define an association rule [Ran97]:

Let us consider $I = \{I_1, I_2, ..., I_m\}$, a set of attributes called items. Let us consider $X \subseteq I$, a subset also called an itemset. An association rule can be defined as an implication:

$$X \Rightarrow Y, where X \subset I, Y \subset I, and X \cap Y = \emptyset$$

In the definition of the association rule, X is the antecedent and and Y is the consequent of the rule.

We can also explain two important concepts for association rule, namely support and confidence. Let the database $D = \{T_1, T_2, ..., T_n\}$ be a multi-set of transactions, where each transaction $T_i, i \in \{1, ..., n\}$, is an itemset.

The support s of an itemset is the fraction of transactions in the database D containing X:

$$s\ (\ X\ ) = \frac{|\{\ T \in D \mid X \subseteq T\ \}|}{|\ D\ |}$$

In accordance to [Zaï99], the support of an association rule is the probability of the two subsets union:

$$Support(X \Rightarrow Y) = Probability(X \cup Y)$$

The rule $X \Rightarrow Y$ holds with confidence c if c is the fraction of transactions X and Y:

$$c\ (\ X\ ,\ \ Y\ ) = \frac{s\ (\ X \cup Y\ )}{s\ (\ X\ )}$$

According to [Zaï99], the confidence of an association rule is the probability of the difference between two subsets:

$$Confidence(X \Rightarrow Y) = Probability(Y \setminus X)$$

A specialization of association rules for text mining will be presented in the chapter concerning specific text mining techniques.

Sequence discovery can be considered more or less like association. Indeed, an association algorithm produces some rules that explain how often occurrences have happened. Nonetheless, contrary to association, in the sequence discovery, time sequence of events is considered. An example to illustrate sequence discovery: Thirty percent of the people who buy a Video Cassette Recorder buy also a camcorder within three months.

Predictive methods

After descriptive methods, we are going to explain predictive methods, namely Classification, Regression and Time Series Analysis.

Classification is a method that classifies a data item into one predefined class. If we take back our example on the loan, on the figure 1.5, we can see an example of classification. In this example, we can see a repartition into class (no loan, loan).

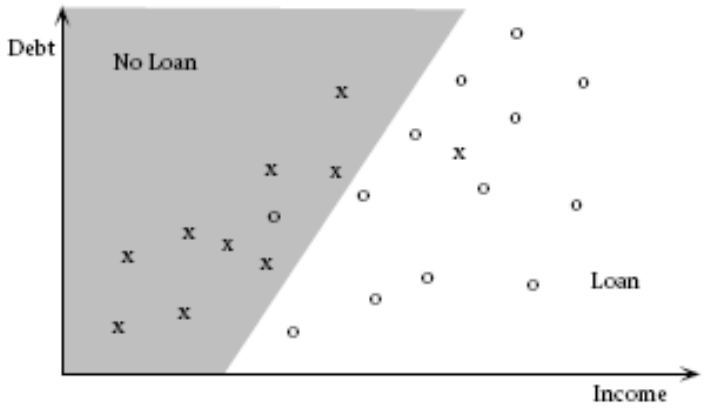

Figure 1.5: Classification applied on the loan example. [FPSS96]

About this example and the classification in general, it is difficult and even impossible to have a neat boundary between two classes because it is not possible to sunder perfectly the classes.

Another predictive method is ***regression***. Regression can be assimilated to a function that map a data item to a prediction variable. Generally, this prediction variable is a real. This method uses existing values to predict what other values will be. It exists two kinds of case.

Firstly, the simplest case that can be represented by a linear regression. On the figure 1.6, a simple case is represented and we can see a regression line.

On the other hand, namely in the more complicated cases, we do not use linear projection due to the fact that there are multiple predictor variables. In these cases, other methods must be used like logistic regression, decision trees or neural

nets.

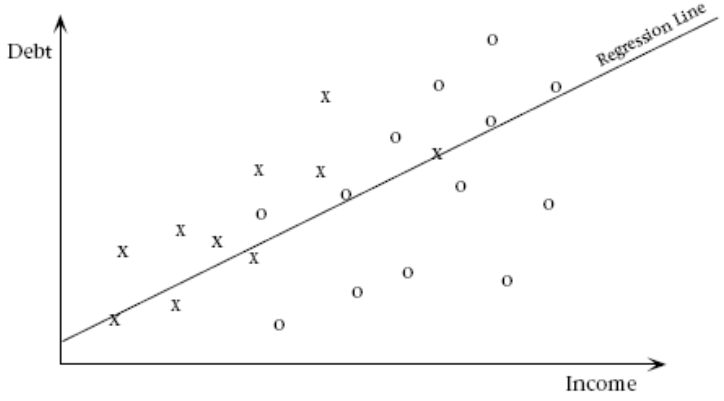

Figure 1.6: Example of the regression method. [FPSS96]

Time series analysis consists of the analyze of time series. A time series is a sequence of observations and these observations are classified in time. In the time series analysis, two variables are generally taken into account: time (the independent variable) and the variable that we analyze (the dependant variable). In the time series, there are two kinds of time series data, namely *continuous* where observations are accomplished at every instant of time (in medicine, electrocardiograms) and *discrete* where observations are regular (in economics, weekly share prices or monthly profits).

1.2.3 The components of Data Mining Algorithms

The methods that we had explained are generally implemented by an algorithm. It will be interesting to analyze the components of a data mining algorithm [FPSS96]. The components are three of them, namely: model representation, model evaluation and search method. We are going explain these terms in details.

Model representation is used by the discoverable patterns. Indeed, the discoverable patterns must be described by a particular language. That is the model representation that is used as language for the discoverable patterns.

Model-evaluation criteria informs (by way of a quantitative statements) if and how a discovered pattern corresponds to the predefined goals of the Knowledge Discovery in Database process.

The last component ***Search method*** is composed of two subcomponents, namely *parameter search* and *model search*. It is required to have realized the model representation and model-evaluation criteria before the search method. Thus, when these two methods are carried out(the data mining problem is reduced), an optimization task can be launched. During this task, the goal is to optimize the evaluation criteria. This optimization is realized by finding the best parameters and models.

In practice, during the first task (parameter search), the best parameters are researched in order that these parameters optimize the model-evaluation criteria. About the second step of this component, this step is a loop applied on the parameter-search method. With this loop, the model representation is altered and from this change, a family of models is taken into account.

This chapter has permitted to present the Data Mining process and Data Mining methods. Now, we are going to deepen our knowledge of the Data Mining with the presentation of the Text Mining characteristics in the following chapter.

Chapter 2

Text Mining: Data Mining Extension

Text Mining (TM), also called Knowledge Discovery from Textual databases (KDT) refers to data mining using text documents as data. It is the process of discovering interesting patterns in a corpus of unstructured data. Text mining combines data mining methods with information extraction techniques, natural language processing, database technology and more. This extension of data mining is thus a more complex task that data mining itself.

Due to the continuous growth of the volume of electronic data currently available, text mining has a big commercial potential. In effect, human are not designed to deal with such an amount of electronic data. Therefore, automated knowledge discovery techniques are on the cynosure.

As text mining deals with unstructured and fuzzy text data, a preliminary step has to be done: the text refining that transforms the first text data input into a intermediate data set that permits to make the first text data suitable for mining.

2.1 Text Mining Process

Different text mining frameworks have already been defined. Text mining could be visualized as consisting of two phases ([Tan99]): *text refining* that permits a data representation which can be automatically treated by a computer and *knowledge distillation* that deduces patterns or knowledge from the refined data.

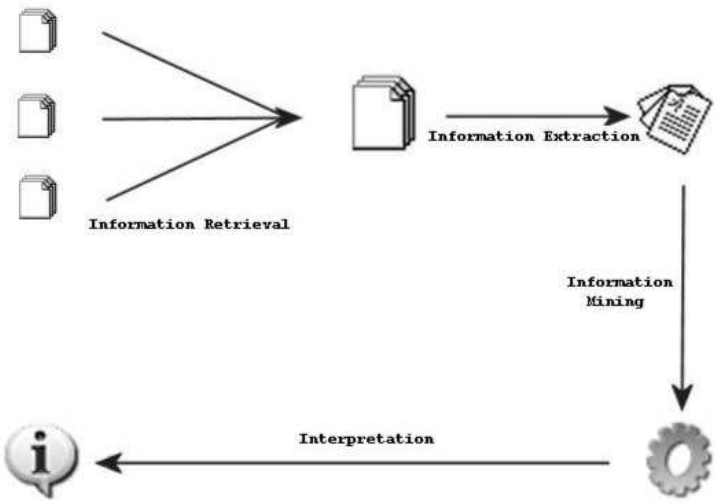

Figure 2.1: Text Mining Framework([Dix97])

The framework considered in that document has been described by [Dix97]. Highly related to the way human do receive and interpret information, text mining can be considered as a four steps process (Figure 2.1):

1. **Information retrieval**: This first step consists in locate and retrieve the documents that can be considered as relevant, i.e. documents that are linked to the considered subject. We unconsciously make that kind of information selection on and on. This step has to be done at hand by users of the system. A lot of information sets pre-classified in terms of subject can be found on the internet databases. Obviously, this first classification is very rough and another system is still needed for a better classification of relevant and irrelevant documents.

2. **Information extraction**: This stage is certainly the most important one. Indeed, a lacunary information extraction would compromise the all text mining process and alter the results. Most of papers concerning text mining treat the information retrieval and the information extraction as the same process, though the are different. This step concerns with scanning a set of documents and extract the facts present in the documents. The user of the system specifies the templates of expected information. Opposite to the

DECLARATION_TYPE	accident
DATE	2002_04_02
NAME	bryant
SURNAME	john
ADRESS	bay street, 2
CITY	sydney
AGE	22
VEHICLE	ford

Figure 2.2: Example of Template

retrieval of information, which is mainly a manual task, the extraction can be computer automated. A most complete description of the information retrieval process will be presented shortly.

3. **Information mining**: Once all documents have been scanned and a template entry has been filled out for each of them, we are in a stage where we have a data structure which is compatible with standard data mining techniques. Thus we can start seeking to discover patterns within the data. For example, you can find on the figure 2.2 a template related to an insurance declaration. The mining of a set of that kind of template could lead to conclusions such as: "People of less than 27 years old have more accident than people of more than 40 years old". Different text mining techniques will be described later on.

4. **Interpretations**: The final step is to place an interpretation on the patterns retrieved from the precedent stage. The interpretation, automated or not, can also be used has a prediction tool. For example, the fact a person bought a newspaper six Mondays at a time could lead to the prediction he will buy another one next Monday! Despite the simplicity of that example, it is easy to realize how interesting it can be to create such a prediction tool. This ability of discovering new information provides a competitive tool for a company that can be used to take advantage on her concurrent.

2.2 Information Extraction

Information extraction is the most important component of the data mining process. Indeed, it has to deal with the "sense" of the documents. The system must be able

to determine if two different texts treat or not on the same subject.

The information extraction process has three major phases:

1. The system extracts individual facts from the document by using local text analysis
2. The system integrates those facts, producing new, larger facts
3. The pertinent facts are translated into the required output format (so that the application of text mining techniques is possible)

[Gri97] gave a large description of the different stages of information extraction process: first, the individual facts are extracted by creating a set of patterns[1] to match the possible linguistic realizations of the facts (*Pattern Matching*). The natural language is too complex to describe these patterns directly as word sequences. Therefore we first begin by structuring the input, identifying various levels of constituents and relations, and then determine our patterns in terms of these constituents and relations. We first start that process with *Lexical Analysis*[2] and *Name Recognition*[3] which are discussed below. This is followed by a *Syntactic Analysis* and after that we use task-specific patterns (*Scenario Pattern Matching*) to identify the facts related to the specific subject.

We come now on the integration phase where the facts from the entire document will be examined and combined. The relations of coreferences will be resolved by a *Coreference Analysis*. It could also be interesting to draw inferences from the different facts in the document. Therefore, an *Inference Analysis* will be needed.

You can find on the figure 2.3 the different stages of the information extraction process.

[1] Lowest level of abstraction

[2] Lexical analysis is used to assign parts-of-speech and features to words and phrases through morphological analysis and dictionary lookup

[3] Name recognition is used to identify names and other special lexical structures such as dates, countries and currency expressions.

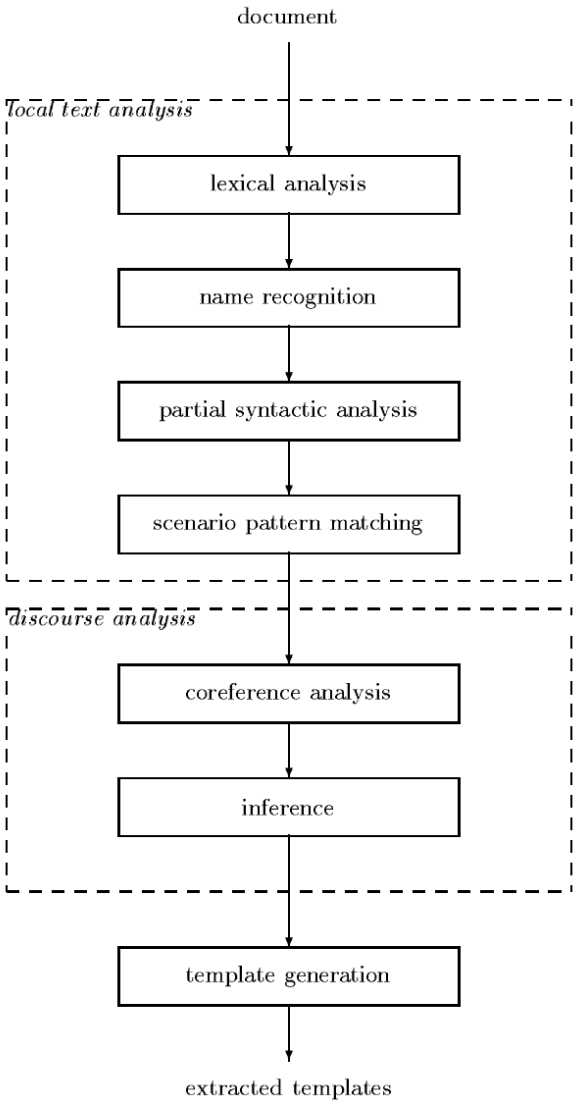

Figure 2.3: Information Extraction Process (Taken from [Gri97])

2.2.1 Pattern Matching

Most of the text analysis is performed by matching the text against a set of regular expressions (patterns). If a certain expression matches a segment of the text, this text segment, called "constituent" is assigned to a label. One approach is to develop rules of information extraction by a manual encoding of the patterns that identify the desired entities or relations.

But due to the variety of forms and contexts in which information can appear, to manually develop patterns is very difficult and not efficient (rarely results in robust systems). Therefore, supervised machine-learning methods trained by a human intervention has become the most successful approach in information extraction. [Mus99] identified three main categories of patterns generated by machine learning algorithms:

Information Extraction from Free Text : the extraction patterns described here are used only to process documents that contain grammatical plain text. The extraction rules to identify the relevant information in a a document are thus based on syntactic and semantic constraints. Therefore, it will be needed to pre-process the original text with a syntactic analyzer and a semantic tagger. For example, AutoSlog ([Ril93] is a system that builds a dictionary of extraction patterns (called *concepts*)). Each concept has

1. a *conceptual anchor*: it is a triggering word that activates the concept
2. a *linguistic pattern* and a set of *enabling conditions* (constraints on the components of the linguistic pattern), which, together guarantee the applicability of the concept.

For example, let us have a look on how the following sentence could be analyzed by AutoSlog if the goal was to know *who has been killed*:

`"Melissa Schroeder has been killed yesterday in the subway"`

We could use the concept that consists on the conceptual anchor "killed" together with the linguistic pattern `<subject> passive-verb`. Here is what would happen if that pattern was applied on the sentence: first, the concept is activated because the sentence contains the triggering word *killed*; then the linguistic pattern is matched against the sentence and the `subject` is extracted as *the person who has been killed.*

Information Extraction from Online Documents : with the expansion of the Web, a huge amount of documents are now accessible for the internet users. Despite those documents are text documents, the information extraction techniques for free text are not fit for those. Indeed, it is necessary to combine syntactic and semantic constraints with *delimiters* that bind the text to be extracted. For example, Whisk ([Sod98]), a learning system that generates extraction rules for a wide variety of documents, uses extraction patterns that have two components: one describes the context that makes a sentence relevant and the other gives the delimiters of the sentence to be extracted. Considering small sale advertisements for cars, here is typical Whisk extraction rule:

```
*'YR'(<Num1>)*'$'(<Num2>)
 → output = YEAR:<Num1> - PRICE:<Num2> US Dollars
```

This rule means: "Ignore all the characters in the text until you find a 'YR' sign immediately followed by a number. Extract that number and fill the "YEAR:" slot with it. Then ignore all the characters in the text until you find a $ sign immediately followed by a number. Extract that number and fill the "PRICE:" slot with it. If we apply this rule to the following advertisement:

```
FORD Mustang
YR 1995
$ 40,000
KM 200,000
E-mail:gertri@gb.com
```

the output will be: `YEAR:1995 - PRICE:40,000 US Dollars`

Wrapper Induction Systems : those systems appear to be necessary when extracting and integrating data from multiple Web-based sources. A typical wrapper application extracts the data from HTML web pages. The wrapper induction systems generate rules that use delimiters but do not use linguistic constraints. For example, Wien [KWD97], which was the first wrapper induction system, generates extraction rules similar to those of Whisk, except that it only uses delimiters that directly precede or follow the considered data. In a Wien system, there is only one (multi-slot) rule that can be used

for all documents. Let us consider that Wien rule used for extracting the name of an hotel and his location:

```
*'.'(*)':'*'('(*)')'
 → output = Hotel {Name @1} {Location @2}
```

This rule means: "Ignore all the characters until you find a '.', sign and extract the hotel's name as the string that ends at the first ':' sign. Then, again, ignore all characters until you find a '(' sign and extract the string that ends at the ')' sign. If we apply this rule to the following document:

```
D1:  num8273.Mercury:Broadway(Sydney)
```

the output will be: `Hotel {Name Mercury} {Location Sydney}`

It is difficult to give a general view of the pattern matching system used for information extraction. Indeed, the way it is realized differs from one system to another. For example, let us have a look on how the extraction system of the New York University Proteus Project (whose the goal is to build a system that can automatically find the information the user is looking for, in his preferred language, at the right level of details) works.

In the NYU extraction system, each pattern has an associated set of actions. The main action generally is the tagging of a text segment with a new label, but other actions may also be performed. The patterns sets are applied one at a time. All the patterns in a set are matched starting at the first word of the sentence:

- If more than one pattern matches, the one matching the longest segment is chosen and the actions associated with that pattern are executed.
- If more than one pattern matches the longest segment, the first is taken and the actions associated with that pattern are executed.
- If no pattern matched, the patterns are reapplied starting at the next word of the sentence

In the case of a pattern matched and an action labeled a text segment, the patterns are reapplied past the end of that segment. All that process continues until the end of the sentence.

2.2.2 Lexical Analysis

At the lowest level, we are concerned with breaking the text into tokens. Each token is looked up in various dictionaries and sets of domain specific lexicons to determine its possible part-of-speech. Special dictionaries can be used at this stage; dictionaries such as major places name dictionary or major companies name dictionary.

2.2.3 Name Recognition

Name recognition is the continuation of the lexical analysis. Proper names and other special forms (such as dates or currency amounts) will now be identified by the system. This step permits to simplify further processing of the data. A set of patterns is used for each of those "special item" categories. For example, personal names can appear in different ways in a document:

`Mr. John Danigton`
→ identified by a preceding title

`John Clark`
→ identified by a common first name

`Candy Smith Jr.`
→ identified by a suffix

`Topper F. Josh`
→ identified by a middle initial

A pattern will be build for each of the forms above. In the same way, company names can be "discovered" using different patterns. Notice that company names can usually be identified by their final token, such as:

`Apple Computer Inc.`

`Microsoft Corporation`

`Hewitt Associates`

It is also possible to find a company name in a document that do not fit with one of the forms mentioned above. Therefore we also need a company names dictionary.

The process of name identification can also include a system that permits to to identify aliases of a name. The system must be able to determine that *John Smith* represents the same person as *Mr. Smith*. This alias recognition can also help the name classification. For example, the system could not know if *Triger Jeff* is a person or a company. But if later in the document we can find *Mr. Triger*, the system will automatically classify Triger Jeff as a person.

2.2.4 Syntactic Analysis

This phase will simplify the subsequent phase of fact extraction. The arguments to be extracted often correspond to phrases in the text when the relationships to be extracted often correspond to grammatical relations. The identification of the complete syntactic structure of a sentence can sometimes be difficult. Thus, depending on the considered system, the amount of syntactic structure which is explicitly identified differs. Some do not even have a syntactic analysis phase. Opposite, some attempt to build a complete parse of the sentence. But in general, most of the systems build structures about which he can be quite certain, from either syntactic and semantic evidence.

This step will thus assign a syntactic component to words (or phrases) in each sentence. Once the basic syntactic of the sentence will be discovered, it will be possible for the information extraction system to look for semantic patterns he has been trained for.

2.2.5 Scenario Pattern Matching

The previous phases have been in a sense preparatory for the scenario pattern matching. The role of this step is to extract the events or relationships relevant to the scenario. For example, if the system analyzes the phrase

```
John Smith killed Mary Bryant
```

where *John Smith* and *Mary Bryant* are patterns that match with the type "Per-

son Name" and *killed* a pattern that match with the type "Verb"(the two pattern types, are normally the results of the Syntactic Analysis but have been here arbitrary chosen), the semantic pattern matcher will associate this sentence with the syntactic structure

personA killed *personB*

and the system will understand that *John Smith* is the murderer of *Mary Bryant*.

2.2.6 Coreference Analysis

The task of coreference analyze is to resolve references by pronouns and definite noun phrase. For example, considering the following sentences:

```
Fred Deban has been killed yesterday.  He was waiting for the train.
```

The coreference analysis will allow the system to determine that the token *He* in the second sentence refers to the token *Fred Deban* in the first sentence.

2.2.7 Inference Analysis

Sometimes, the information concerning an event may be spread over several sentences. The system has to combine the information before a template can be generated. Moreover, the information can also be implicit and an inference process is needed to make it explicit. For example, let us imagine the following phrases are found in the text:

```
John will be minister; He succeeds Garby
```

With a inference analysis, the system will determine that Garby was minister. Such inferences can be implemented using production rules. Notice that the example above did not require to take account of the time of each event. However, it can sometimes be important to take account of the time.

Once the relevant facts have been extracted in a format (one template for each document) compatible with the standard data mining techniques, the text mining

can be applied.

2.3 Information Mining

Most of the text mining techniques are based on data mining techniques. We will here describe the principal text mining techniques which are, for some, purely data mining techniques.

2.3.1 Episodes and Episode Rules

Episode and episodes rule are an extension of the data mining concepts of *association rules* and *frequent sets*.

Text Episodes

[AHKV98] showed that sequential data, especially text data can be seen as a sequence of tuples:

`(feature_vector,index)`*

with *feature vector* a set of features; a feature can be

- a word
- a phrase
- a punctuation mark
- a mark-up tag

and *index* an element containing information about the position of the word in the sequence.

After the definition of *feature vector*, we are now able to define a *text episode*: It is a pair α so that

$$\alpha = (V, \leq)$$

with V a collection of *feature vectors* and $\leq$ a partial order on V. Considering a text sequence S, we say that a text episode $\alpha = (V, \leq)$ *occurs* within S if the

feature vectors of V can be found in S, so that the partial order $\leq$ is satisfied.

We still have to find a way to determine if an occurrence of the episode is interesting or not. For an episode to be interesting, all his feature vectors must occur close enough in S. To determine if it is "close enough", we give a limit, called the *window size* (W), within which the episode must occur. Thus, the examination work of the occurrences is reduced to the substrings S' of S where the difference of the indices of the feature vectors in S' is at most W. Let us consider that text sequence:

```
Federal Reserve pleased with inflation progress
```

Considering that the feature vector contains the base form of words and the number of each word, the text sequence will be represented by the following sequence of tuples:

```
{(federal,1);(reserve,2);(please,3);(with,4);(inflation,5);(progress,6)}
```

For a window size (M) of 2, this sequence contains the episode

`(federal, reserve)`, and with the indices: `(federal`^{1}`, reserve`^{2}`)`)

because the difference between the two indices is: (2-1) $\leq M$. On the other hand, the sequence does not contain the episode

`(federal, progress)`, and with the indices: `(federal`^{1}`, progress`^{6}`)`)

because the difference between the two indices is: (6-1) $\not\leq M$.

Episode Rules

An episode rule gives the conditional probability that a certain episode occurs (in an interval of given window size (W)), given that a subepisode has occurred (in an interval of given window size ($V : V \leq W$)). An episode rule is thus an expression of the type:

$$\beta[V] \Rightarrow \alpha[W]$$

$$W \leq V$$

The *confidence* of the rule is the conditional probability that α occurs, given that β has occurred, under window size constraints specified by the rule. Let us analyze an example of episode rule:

`information, extraction, from [4]` $\Rightarrow$ `databases [5] (76%)`

This rule has the following meaning: when the words {`information, extraction, from`} occur in an interval of 4 consequent words, there is a probability of 0.76 that the words {`information, extraction, from, databases`} occur in an interval of 5 consequent words. The rules with a negligible confidence are not interesting. Therefore, usually, the selected rules are those with a confidence exceeding a given level of confidence, called the *confidence threshold*.

Postprocessing the results

Now that a large amount of episodes and episode rules has been produced, the postprocessing phase is needed to allow to focus our study. The main point is now to define which episodes have to be considered. The relevance of episodes is highly dependent on the kind of application, but there are measures that are common in all documents, independent of the semantic content of the data texts. A way to enhance the usability of the results is to use knowledge on the rules of a single document, and then compare it to similar information on the entire document. Here are some measures based on the rules of one document:

1. **Length**: The absolute length ($Length$) of an episode is the number of feature vectors it contains. The length of a rule is the length of its episode. To scale the length, we compare it to the maximum window size ($WinSize_{max}$) and we find the relative length ($Length_{rel}$):

$$Length_{rel} = \frac{Length}{WinSize_{max}}$$

   For example, let us consider $WinSize_{max}$= 4, the maximum window size and the following episodes (which contains 2 and 3 features vector respectively):

   `(federal, reserve)`

   `(federal, reserve, please)`

the relative lengths are respectively,

$$Length_{rel} = \frac{2}{4} = 0,5$$

$$Length_{rel} = \frac{3}{4} = 0,75$$

$\Rightarrow$ it is interesting to take the *length* into account because longer phrases should be preferred. Indeed, they are more descriptive and it is always possible to reconstruct shorter phrases from longer one.

2. **Tightness**: Considering $Length_1$ the length of the left-hand side of the rule, $Length_2$ the length of the entire rule and $WinSize_1$, $WinSize_2$ the window sizes of the left-hand side and the entire rule, respectively. We denote by D the mean of the differences of these measures:

$$D = \frac{(WinSize_1 - Length_1) + (WinSize_2 - Length_2)}{2}$$

The *tightness* of the rule is

$$Tightness = 1 - \frac{D}{WinSize_{max}}$$

$\Rightarrow$ The length of a phrase is limited by the window size. If we want to see long phrases, we have to increase the window size. But this could also result in short phrases spreading out too much. To counteract that effect, we use *tightness* that gives a way to decrease the weight of these possible undesired phrases.

3. **Mutual Confidence**: For that measure, we need to define two new concepts: the *minimal occurrence* and the *support*.

   $\rightarrow$ Let t, t', u, u' four real numbers such as $t \leq u \leq u' \leq t'$. An occurrence of a text episode (α) at the interval $[t, t']$ is minimal if α does not occur in any proper subinterval $[u, u'] \subset [t, t']$ (from [MT96]).

   $\rightarrow$ The *support* of a text episode (α) in a text sequence (S) is defined as the number of minimal occurrences of α in S.

   For example, let us consider the following episode:

   `(federal, reserve)`,

and the following text:

The American Federal Reserve said the inflation is progressing. The Federal Reserve released a new report today.

→ The support of the episode equals 2.

Let $Support_{left}$, $Support_{right}$ and $Support_{all}$ the supports of the left-hand side, the right-hand side, and the entire rule, respectively. The *mutual confidence* (M) of the rule is

$$M = \frac{\frac{Support_{all}}{Support_{left}} + \frac{Support_{all}}{Support_{right}}}{2}$$

⇒ Mutual confidence is used to reveal ties between the left-hand side and the right-hand side of the rule. This formula prefers cases in which words appear often together and seldom with some other words.

Conclusion

Episode and Episode Rules are an interesting concept in knowledge discovery. Nevertheless, in our case, due to the quantity of text information to be processed, the Episode Rules technique would be too time-consuming. This method is efficient for relatively small data sets.

2.3.2 Conceptual Clustering

Clustering is a well-known data mining technique. As this technique has already been discussed in this document, we just show you here how it can be used in text mining. In text mining, clustering will look at building up clusters of documents that are related. The goal of the algorithm working on the text will be to determine clusters with a minimal intra-cluster distance and a maximal inter-cluster distance.

Most of clustering techniques are concept dependent. Indeed, most of domain dependent variables are hand coded by a domain expert. An important point would be to generate automatically such information (by parsing the text) so that there will be no need anymore to code it by hand. But this issue of a generic clustering is not new, and some work has already been done in that way (for example [BB99]).

Conclusion

Clustering is a global concept containing lots of different techniques. Indeed, the field of clustering has undergone major revolution over the last few decades and we now have several algorithms using clustering (for example algorithms for clustering massively large data sets). Thus, the advantage of clustering techniques is that they can be applied in very different contexts. They are highly adaptable. In our case, the technique we use for the prediction phase is based on clustering: Support Vector clustering.

2.3.3 Concept Hierarchies

This approach has been discussed by [FD95]. It use concepts and their hierarchies. The goal is to organize a set of documents into a concept hierarchy, where an overall structure of them can be observed. Feldman points out that the current limitations of natural language processing show that we cannot be as sophisticated with this process as we would like to be. Indeed, the most a method is sophisticated the most she is time consuming. With the amount of data to be processed, we cannot afford too sophisticated methods. Therefore the process of concept hierarchy is interesting. The method is simple enough to be fast and still give us enough structure with which we still can find interesting facts.

The concept hierarchy is represented by a graph of concepts relationships where a *parent-child* relationship indicates that the parent is more general concept than the child. For example, on the figure 2.4 a concept hierarchy concerning a color classification is represented. We can see that "red" belong to the larger concept "Monochrome color" which belongs to the larger concept "Color". The hierarchy can be customized so that it only contains concepts and relationships that may be relevant for the user. For that customization we need a domain-specific structure that is usually realized by hand.

Once the concept hierarchy has been established, we use it to tag words or phrases contained in the document. Each word (or phrase) is tagged with the relevant concept and, implicitly, all ancestors of the considered concept are tagged too. Now that the sets of concepts contained in each document have been generated, we can begin to apply data mining techniques looking at relationships within and

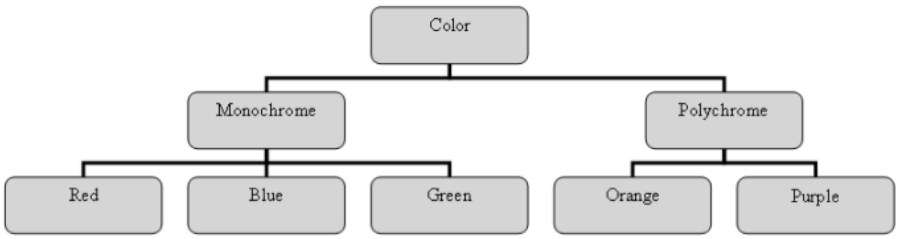

Figure 2.4: Example of Concept Hierarchy

between the documents.

Conclusion

The Concept hierarchy method presents a good compromise between a full syntactic analysis and no syntactic analysis. Nevertheless, certainly due to the huge data set we have in our project, this method is not used. Counter to [DLTV05], we do not use any syntactic analysis that would be too time-consuming due to the size of our data set.

2.3.4 Neural Network approach

The following explanations come from [SS].

Introduction

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between neurones. This is true for ANNs as well.

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions.

Neural networks take a different approach to solve problem then conventional computers. Conventional computers use an algorithmic approach i.e. the computer follows a set of instructions in order to solve a problem. Unless the specific steps that the computer needs to follow are known the computer cannot solve the problem. That restricts the problem solving capability of conventional computers to problems that we already understand and know how to solve. Neural networks process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements(neurones) working in parallel to solve a specific problem. Neural networks learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly. The disadvantage of that system is that since the network finds out how to solve the problem by itself, its operation can be unpredictable.

Neurons and firing rules

An artificial neuron is a device with many inputs and one output that "imitates" the behavior of a neurone (fig. 2.5). The neuron has two modes of operation; the training mode and the using mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong to the taught list of input patterns, the firing rule is used to determine whether to fire or not.

The firing rule is an important concept in neural networks and accounts for their high flexibility. A firing rule determines how we calculate whether a neuron should fire for any input pattern. It relates to all input patterns, not only the ones on which the node was trained. For example, a 3-input neuron is taught to output 1 when the input (X1,X2 and X3) is 111 or 101 and to output 0 when the in-

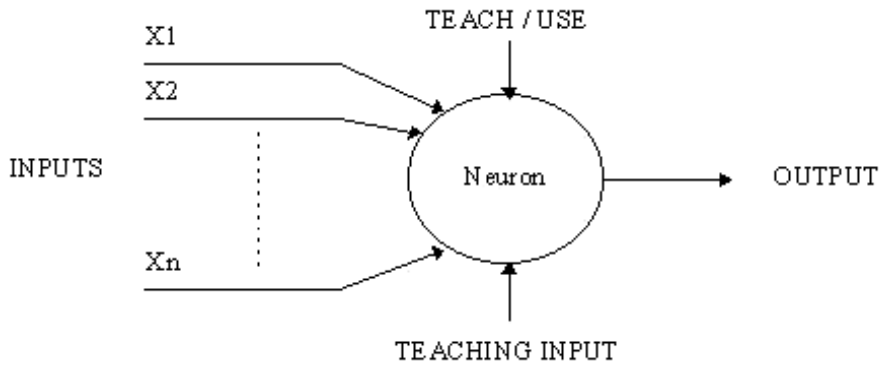

Figure 2.5: A simple Neuron

put is 000 or 001. Then, before applying the firing rule, we have the inital table 2.1.

X1:	0	0	0	0	1	1	1	1
X2:	0	0	1	1	0	0	1	1
X3:	0	1	0	1	0	1	0	1
OUT:	0	0	0/1	0/1	0/1	1	0/1	1

Table 2.1: Initial Firing Table

As an example of the way the firing rule is applied, take the pattern 010. It differs from 000 in 1 element, from 001 in 2 elements, from 101 in 3 elements and from 111 in 2 elements. Therefore, the "nearest" pattern is 000 which belongs in the 0-taught set. Thus the firing rule requires that the neuron should not fire when the input is 001. On the other hand, 011 is equally distant from two taught patterns that have different outputs and thus the output stays undefined (0/1). Thus, by applying the firing in every column, the initial truth table is transformed into the table 2.2.

X1:	0	0	0	0	1	1	1	1
X2:	0	0	1	1	0	0	1	1
X3:	0	1	0	1	0	1	0	1
OUT:	0	0	0	0/1	0/1	1	1	1

Table 2.2: Modified Firing Table

The difference between the two truth tables is called the *generalization of the neuron*. Therefore the firing rule gives the neuron a sense of similarity and enables it to respond 'sensibly' to patterns not seen during training.

MCP Neuron

The previous neuron does not do anything that conventional computers do not do already. A more sophisticated neuron (2.6) is the McCulloch and Pitts model (MCP). The difference from the previous model is that inputs are "weighted", the effect that each input has at decision making is dependent on the weight of the particular input. The weight of an input is a number which when multiplied with the input gives the weighted input. These weighted inputs are then added together and if they exceed a pre-set threshold value, the neuron fires. In any other case, the neuron does not fire.

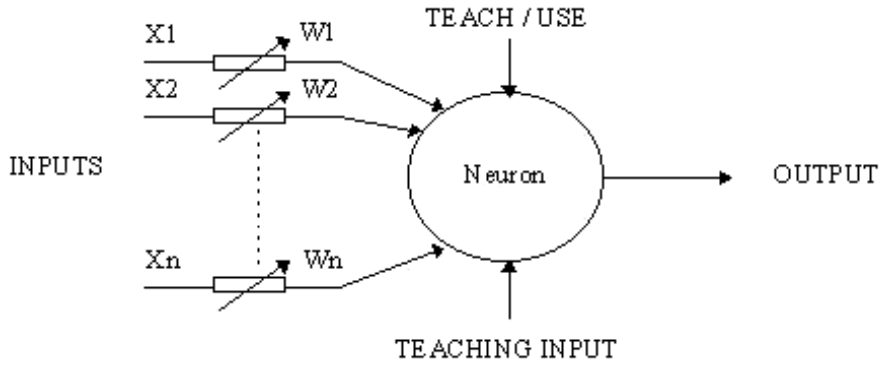

Figure 2.6: An MCP Neuron

In mathematical terms, the neuron fires if and only if

$$X1W1 + X2W2 + X3W3 + ... > T$$

thus

$$\sum_{i=1}^{n} X_i W_i > T$$

The introduction of input weights and of a threshold makes this neuron a very flexible and powerful one. The MCP neuron has the ability to adapt to a particular situation by changing its weights and/or threshold. Various algorithms exist that cause the neuron to "adapt" himself.

Neural Networks Architecture

There are two types of Neural Networks architecture:

Feed-forward networks Feed-forward ANNs (fig. 2.7) allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organization is also referred to as bottom-up or top-down.

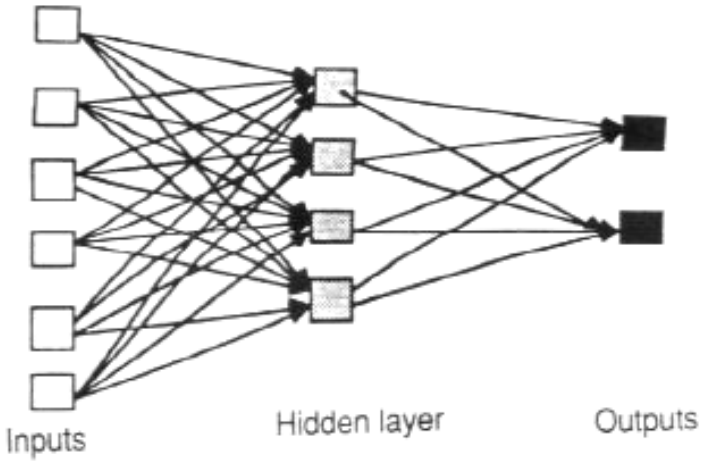

Figure 2.7: Feed-forward network

Feedback networks Feedback networks (fig. 2.8) can have signals traveling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic; their "state" is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organizations.

Perceptrons

The most influential work on neural networks in the years 60 went under the heading of "perceptrons" a term coined by Frank Rosenblatt. The perceptron (fig.

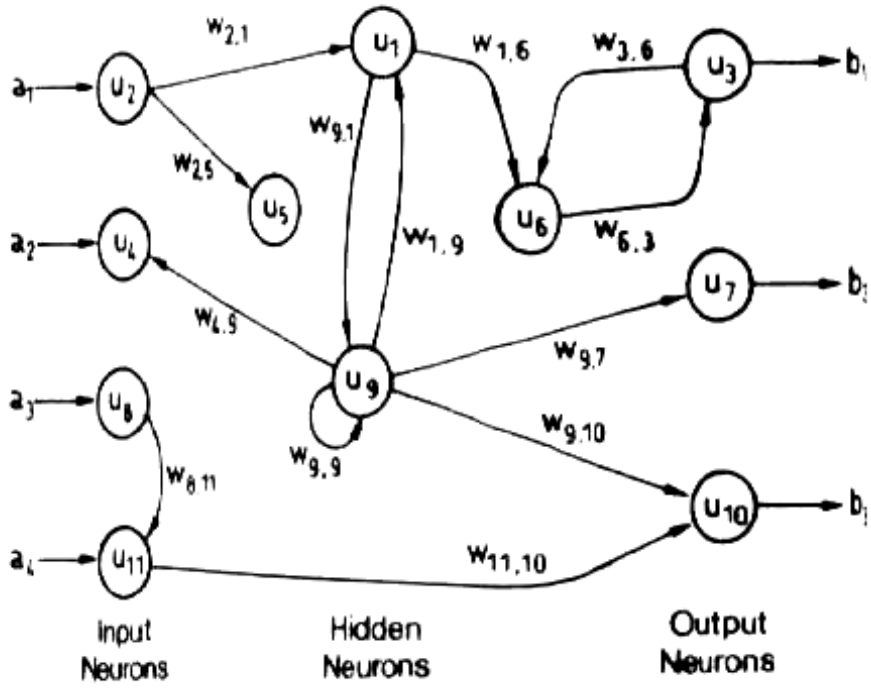

Figure 2.8: Feedback Network

2.9) turns out to be an MCP model (neuron with weighted inputs). Units labeled A1, A2,..., Aj ,..., Ap are called association units and their task is to extract specific, localized features from the input images. Perceptrons imitate the basic idea behind the mammalian visual system. They were mainly used in pattern recognition even though their capabilities extended much more.

Conclusion

Neural Networks is an interesting approach. We have significant results especially in the imagery domain. The main advantage of neural networks is that they are very adaptive and they can easily evolve depending on the new input they get. Despite those facts, we do not use that technique but it could be interesting to analyze if it could bring us best results.

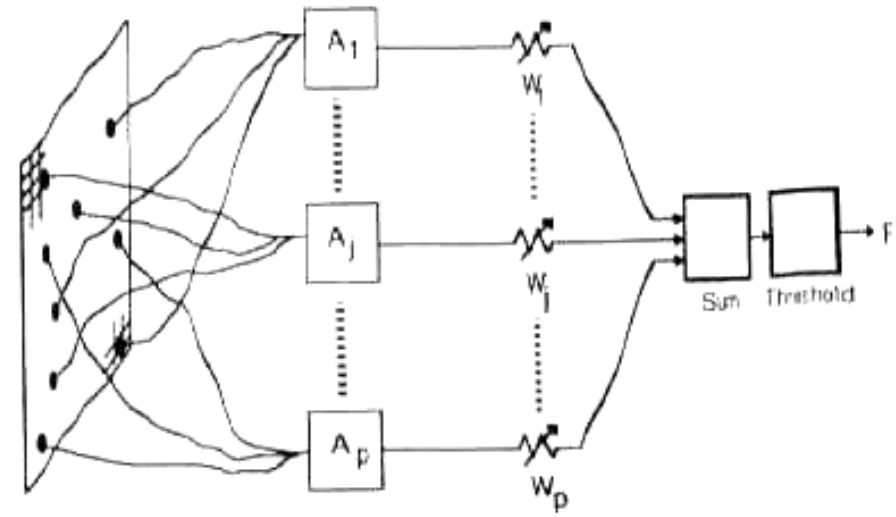

Figure 2.9: Perceptron

Chapter 3

Support Vector Machines Prediction

This section is mainly based on [Bur98], [KRMS01], [Mah03]. The algorithm of Support Vector Machines has been developed in the nineties by Vapnik as a supervised binary classification algorithm. It performs a classification by constructing an N-dimensional hyperplane that optimally separates the data into two categories. The algorithm is basically designed for finding a linear decision border, but this model can also be improved by projecting himself in another space so that the divisibility of the data can increase. Then, the same algorithm can be applied so that the decision border in the initial space will not be linear anymore.

The Support Vector Machines Algorithm is well-known in the community of Machine Learning for its good performance and for the fact he finds a unique solution. But as explained earlier, its strength is the projection mechanism that permits switch of space for the learning process. For some spaces, a "kernel function" helps to do this projection. This aspect is described below.

The chapter is divided into two parts: the linear Support Vector Machines (an hyperplane can be found in the initial space to separate data) and the nonlinear Support Vector Machines (the initial space contains no hyperplane able to separate data).

3.1 Linear Support Vector Machines

The task of classification is to find a decision rule, which, based on external observations, assigns an object to one of several classes. In the simplest case, which is

the case used in this project, they are only two different classes. In a formal way, the task is to estimate a function

$$f : \mathbb{R}^N \Rightarrow \{-1, +1\}$$

from a learning set of couples (x_i, y_i), supposed independent identically distributed (i.i.d), according to an unknown probability distribution

$$P(x, y)$$

$$(x_i, y_i) \in \mathbb{R}^N \times Y \text{ with } i = 1, ..., N \text{ and } Y = \{-1, +1\}$$

such that f will correctly classify unseen examples (x_t, y_t). For instance, an example could be assigned to the class $(+1)$ if $f(x_t) \geq 0$ and to the class (-1) otherwise.

3.1.1 Margins

Let us for a moment assume that the training sample is separable by an hyperplane, i.e., we choose decision functions of the form

$$f(x) = \langle w, x \rangle + b \tag{3.1}$$

The figure 3.1 represents a linear classifier and its margins: A linear classifier is defined by a normal vector w and an offset b of the hyperplane, i.e., the decision boundary is $\{x | \langle w, x \rangle + b = 0\}$ (thick line). Each of the two halfspaces defined by this hyperplane corresponds to one class, i.e., $f(x) = sign(\langle w, x \rangle + b)$. The margin of a linear classifier is the minimal distance of any training point to the hyperplane. In this case it is the distance between the dotted lines and the thick line.

As said earlier, the margin is defined as the minimal distance of a sample to the decision surface. The margin in turn can be measured by the length of the weight vector, w in (3.1). As we assumed that the training sample is separable, we can rescale w and b such that the points (x_i) that are closest to the hyperplane satisfy $|\langle w, x_i \rangle + b| = 1$ (i.e., obtain the so-called canonical representation of the hyperplane).

Now consider two samples x_1 and x_2 from different classes with $\langle w, x_1 \rangle + b = 1$ and $\langle w, x_2 \rangle + b = -1$, respectively. Then the margin γ is given by the distance

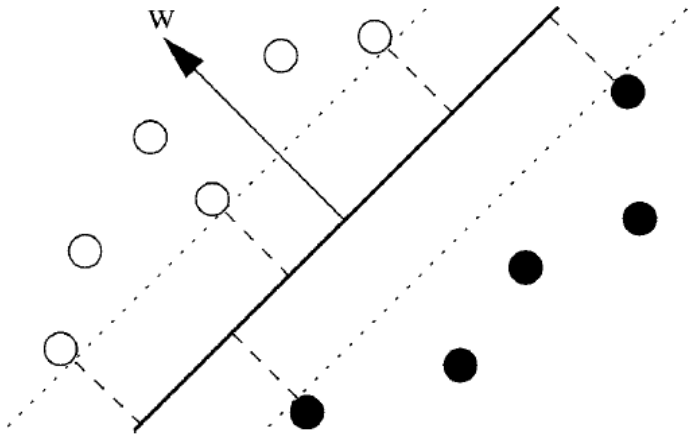

Figure 3.1: Linear Classifier and Margins

between x_1 and x_2, measured perpendicularly to the hyperplane:

$$\gamma = \left\langle \frac{w}{\|w\|}, x_1 - x_2 \right\rangle = \frac{2}{\|w\|}$$

Thus, as the goal is to maximise the margin γ, we have to minimize $||w||$.

The best function f that can be obtained is the one minimizing the expected error (*risk*):

$$R[f] = \int L(f(x), y) dP(x, y), \tag{3.2}$$

where L denotes a suitably chosen loss function, e.g., the *squared loss* which is the most common loss function for regression problems

$$L[f(x), y] = (f(x) - y)^2.$$

Unfortunately the risk (3.2) cannot be minimized directly, since the underlying probability distribution $P(x, y)$ is unknown. Therefore, we have to try to estimate a function that is close to the optimal one based on the available information, i.e., the learning set and properties of the function class F the solution f is chosen from. To this end, we will approximate the minimum of the risk (3.2) by the minimum

of the *empirical risk* which is written

$$R_{emp}[f] = \frac{1}{N}\sum_{i=1}^{N} L[f(x_i), y_i]. \tag{3.3}$$

It is possible to give conditions on the learning machine which ensure that asymptotically (as $N \to \infty$), the empirical risk (3.3) will converge towards the expected risk (3.2). However, if we do not have much learning data (if N is small) the risk of overfitting is present.

Example illustrating overfitting: Assume a set of n documents annotated depending on their category, represented in R^2. A document is thus represented by two reals, for example, the occurrence number of the terms "euro" and "dollar". This is shown on figure 3.2. The positive and negative examples are respectively represented by the signs $\times$ and $\circ$. As shown on figure 3.2 the function does not make any assignation mistake on the documents used for the learning stage (on the left side). On the other side, on new documents (on the right side) we can see that the half of them are bad classified (same performance as an aleatory classifier).

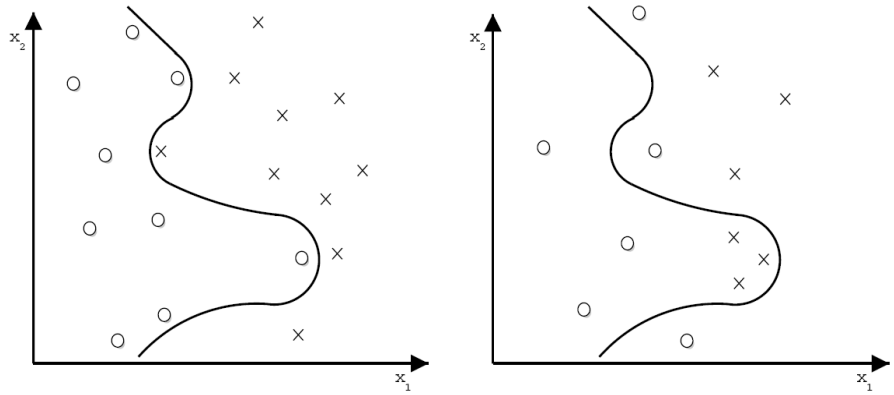

Figure 3.2: Overfitting Illustration

One way to avoid the overfitting dilemma is to restrict the complexity of the function class F which the function f belongs to. The intuition, which will be formalized in the following is that a "simple" (e.g., linear) function that explains most

of the data is preferable to a complex one. Therefore we introduce a regularization term to limit the complexity of the function class F.

3.1.2 Vapnik-Chervonenkis Theory

A specific way to control for the complexity of a function class is given by the Vapnik-Chervonenkis (VC) theory and the structural risk minimization (SRM) principle. Here the concept of complexity is captured by the VC dimension (noted h) of the function class F that f belongs to. Roughly speaking, the VC dimension measures how many items from the training set can be separated by all possible labelings issued from the functions of the class.

Assume a nested family of function classes

$$F_1 \subset ... \subset F_k$$

with nondecreasing VC dimension the structural risk minimization principle proceeds as follows: Let $f_1, ...f_k$ be the solutions of the empirical risk minimization 3.3 in the function classes F_i. SRM chooses the function class F_i (and the function f_i) such that an upper bound on the generalization error is minimized which can be computed making use of theorems such as the following one ([Vap98], [Vap95]).

Theorem 1 *Let h denote the VC dimension of the function class F and let $R_{emp}[f]$ the empirical risk defined by 3.3 using the 0/1 loss function.*[1] *For all $\delta > 0$ and $f \in F$ the inequality bounding the risk*

$$R[f] \leq R_{emp}[f] + \sqrt{\frac{h(ln\frac{2N_x}{h} + 1) - ln(\delta/4)}{N_x}} \tag{3.4}$$

holds with probability of at least $(1 - \delta)$ for $N > h$.

Note that this bound is only an example and similar formulations are available for other loss functions and other complexity measures. The goal here is to minimize

[1] Denote by $(x, y, f(x)) \in X \times Y \times Y$ the triplet consisting of a pattern x, an observation y and a prediction $f(x)$. The 0/1 loss function is the simplest case to consider. It involves counting the misclassification error if pattern x is classified wrongly we incur loss 1, otherwise there is no penalty:

$$c(x, y, f(x)) = \begin{cases} 0 & \text{if } y = f(x) \\ 1 & \text{otherwise} \end{cases}$$

the generalization error $R[f]$, which can be achieved by obtaining a small training error $R_{emp}[f]$ while keeping the function class as small as possible. The inequality 3.4 makes arise two extremes:

1. a very small function class (like F_1) produces a vanishing square root term, but a large training error might remain.

2. a huge function class (like F_k) may give a vanishing empirical error but a large square root term.

The best class is usually between those two extremes, as one would like to obtain a function that explains the data quite well and to have a small risk in obtaining that function.

The figure 3.3 illustrates well the dilemma. The dotted line represents the training error (*empirical risk*), the dash-dotted line the upper bound of the complexity term (*confidence*). With higher complexity the empirical error decreases but the upper bound on the risk confidence becomes worse. For a certain complexity of the function class, the best expected risk (solid line) is obtained. Thus, in practice the goal is to find the best tradeoff between empirical error and complexity.

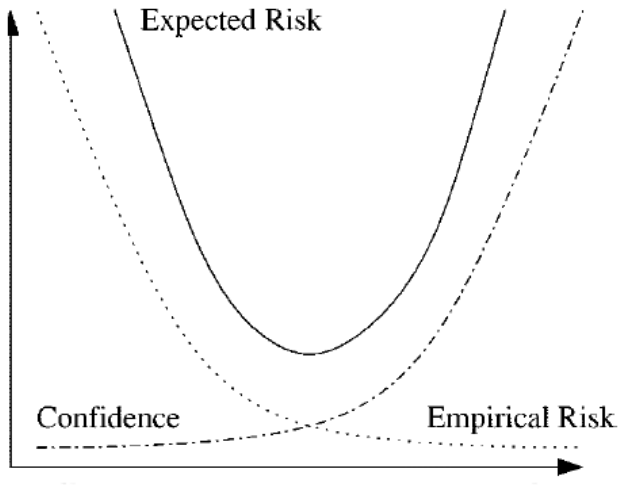

Figure 3.3: Complexity of Function Set ([KRMS01])

One strategy is to keep the empirical risk zero by constraining w and b to the perfect separation case of the samples in the learning set, while minimizing the complexity term, which is a monotonically increasing function (h) of the VC dimension. For a linear classifier the VC dimension h is bounded according to $h \leq \|w\|^2 R^2 + 1$, where R is the radius of the smallest ball around the training data, which is fixed for a given data set. Thus, we can minimize the complexity term by minimizing $\|w\|^2$. This can be formulated as a quadratic optimization problem:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \tag{3.5}$$

3.2 Unlinear Support Vector Machines

Before going further, let us generalize the concept of Support Vector Machines in the case of it is not possible to find an hyperplane to separate the data. The power of Support Vector Machines is that when an hyperplane cannot be found in the initial space, it is possible to map the data in a higher space so that the data can be linearly separated. To map the data, a kernel function is used.

3.2.1 Feature space and kernel functions

The choice of linear functions seems to be very limiting (instead of being likely to overfit we are now more likely to underfit). Fortunately those models can be deeply improved by mapping the data into a potentially much higher dimensional feature space $\mathcal{F}$. The goal of this mapping is for the data set to be linearly separable.

Lets consider the function Φ defined by:

$$\begin{array}{rccc} \Phi: & \mathcal{X} & \rightarrow & \mathcal{F} \\ & x & \mapsto & \Phi(x) \end{array}$$

For a given learning problem, we now consider the same learning algorithm in $\mathcal{F}$ instead of in $\mathcal{X}$, considering the following set:

$$(\phi(x_i), y_i) \in \mathcal{F} \times \mathcal{Y} \text{ where } i = 1, ..., N \text{ et } \mathcal{Y} = \{-1, +1\}$$

The so-called *curse of dimensionality* from statistics says essentially that the difficulty of an estimation problem increases drastically with the dimension N of the

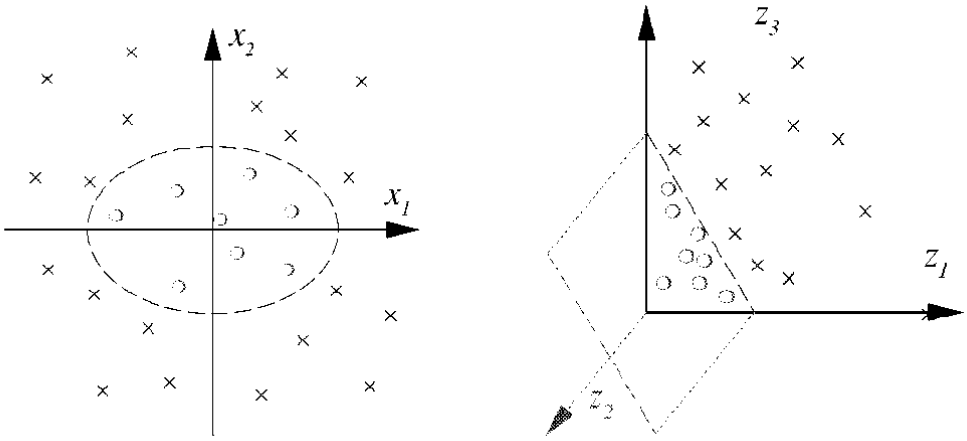

Figure 3.4: Two-dimensional classification example

space, since ,in principle, as a function of N needs exponentially many patterns to sample the space properly. This well-known statement induces some doubts about whether it is a good idea to go to a high-dimensional feature space for learning.

However, statistical learning theory tells us that the contrary can be true: learning in $\mathcal{F}$ can be simpler if one uses a low complexity, i.e., simple class of decision rules. All the variability and richness that we need to have a powerful function class is then introduced by the mapping Φ.

In short: only the complexity of the function class and not the dimensionality matters. Intuitively, this idea can be understood from the example in the figure 3.4. In two dimensions, a rather complicated nonlinear decision surface is necessary to separate the classes, whereas in a feature space of second-order monomials

$$\begin{array}{rccccc} \Phi: & \mathbb{R}^2 & \rightarrow & \mathbb{R}^3 & & \\ & (x_1, x_2) & \mapsto & (z_1, z_2, z_3) & := & (x_1^2, \sqrt{2}x_1x_2, x_2^2) \end{array} \tag{3.6}$$

all we need for separation is a linear hyperplane. In this simple example, we can easily control both: the statistical complexity (by using a simple linear hyperplane classifier) and the algorithmic complexity of the learning machine, as the feature space is only three dimensional. However, it becomes rather tricky to control the latter for large real-world problems.

Fortunately, for certain feature spaces $\mathcal{F}$ and corresponding mappings Φ there is a highly effective trick for computing scalar products in feature spaces using *kernel functions* such as

$$k(x_1, x_2) = \langle \Phi(x_1), \Phi(x_2) \rangle$$

For instance, if we come back to the example from 3.6, the computation of a scalar product between two feature space vectors, can be readily reformulated in terms of a kernel function k:

$$\begin{aligned}(\Phi(x) \cdot \Phi(y)) &= (x_1^2, \sqrt{2}x_1x_2, x_2^2)(y_1^2, \sqrt{2}y_1y_2, y_2^2)^\top \\ &= ((x_1, x_2)(y_1, y_2)^\top)^2 \\ &= (x \cdot y)^2 \\ &= k(x, y)\end{aligned}$$

There are different type of kernel functions, here are some examples:

Linear kernel

$$k(x, x') = x \cdot x'$$

Polynomial kernel

$$k(x, x') = (x \cdot x')^d \text{ or } (c + x \cdot x')^d$$

Gaussian kernel

$$k(x, x') = e^{-\|x-x'\|\frac{2}{\sigma}}$$

Laplacian kernel

$$k(x, x') = e^{-\|x-x'\|\frac{1}{\sigma}}$$

The interesting point about kernel functions is that the scalar product can be implicitly computed in $\mathcal{F}$, without explicitly using or even knowing the mapping Φ. Thus, every linear classification algorithm that can be formalized by scalar products can be extended to the nonlinear classification using a kernel function (chosen a priori).

3.2.2 Back to the minimization

Let us consider again a classifier based on a separating hyperplane. For separating hyperplane classifiers, the conditions for classification without training error are

$$y_i(\langle w, x_i \rangle + b) \geq 1, i = 1, ..., N$$

As linear function classes are often not rich enough in practice, we will consider linear classifiers in feature space using dot products. To this end, we substitute $\Phi(x_i)$ for each training example x_i. Thus, in feature space, the conditions for perfect classification are described as

$$y_i(\langle w, \Phi(x_i) \rangle + b) \geq 1, i = 1, ..., N \tag{3.7}$$

The goal of learning is to find $w \in \mathcal{F}$ and b such that the expected risk (3.2) is minimized. However, since we cannot obtain the expected risk itself, we will minimize the bound of the inequality (3.4), which consists of the empirical risk and the complexity term. We previously showed that a good strategy was to keep the empirical risk zero by constraining w and b to the perfect separation case of the samples in the learning set, while minimizing the complexity term, which is a monotonically increasing function (h) of the VC dimension. For a linear classifier in feature space $\mathcal{F}$ the VC dimension h is bounded according to $h \leq \|w\|^2 R^2 + 1$, where R is the radius of the smallest ball around the training data, which is fixed for a given data set. Thus, we can minimize the complexity term by minimizing $\|w\|^2$. We showed that this can be formulated as a quadratic optimization problem:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \tag{3.8}$$

However, if the only possibility to access the feature space $\mathcal{F}$ is via dot-products computed by the kernel, we cannot solve 3.8 directly since $w \in \mathcal{F}$. But it turns out that we can get rid of the explicit usage of w by forming the dual optimization problem. Introducing Lagrange multipliers $\alpha_i, i = 1, ...N$ one for each of the constraints in (3.7), we get the following Lagrangian:

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i (y_i(\langle w, \Phi(x_i) \rangle + b) - 1) \tag{3.9}$$

(which would be

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i (y_i(\langle w, x_i \rangle + b) - 1) \tag{3.10}$$

in the case of a separating hyperplane can be found in the initial space.)

Remind that introducing positive constraints in the Lagrangian comes to multiply them by positive coefficients ($\lambda_i \geq 0$) and subtract them from the function to be optimized.

The task is to minimize $\mathcal{L}(w, b, \alpha)$ with respect to w, b and to maximize it with respect to α_i. At the optimal point, we have the following saddle point equations:

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \text{ and } \frac{\partial \mathcal{L}}{\partial w} = 0$$

which translate into

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \text{ and } w = \sum_{i=1}^{N} \alpha_i y_i \Phi(x_i) \tag{3.11}$$

From the right equation of (3.11), we find that w is contained in the subspace spanned by the $\Phi(x_i)$. By substituting (3.11) into (3.9) and by replacing $\langle \Phi(x_i), \Phi(x_j) \rangle$ with kernel functions $k(x_i, x_j)$, we get the dual quadratic optimization problem: maximize the Lagrangian with respect to the coefficients α_i under the condition that its saddle point equation equals zero, thus:

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

$$\text{subject to } \begin{cases} \alpha_i \geq 0, i = 1, ...N \\ \sum_{i=1}^{N} \alpha_i y_i = 0 \end{cases}$$

Thus, by solving the dual optimization problem, we obtain the coefficients $\alpha_i, i = 1, ..., N$, that we need to express the vector w which solves (3.8). This leads to the nonlinear decision function

$$f(x_t) = sign \left\{ \sum_{i=1}^{N} y_i \alpha_i \langle \Phi(x_t), \Phi(x_i) \rangle + b \right\}$$

i.e.

$$f(x_t) = sign\left\{\sum_{i=1}^{N} y_i \alpha_i k(x_t, x_i) + b\right\}$$

Note that we have up to now only considered the separable case, which corresponds to an empirical error of zero (cf. Theorem 1). However, the learning data might be noisy and nonseparable, even in the feature space $\mathcal{F}$. Therefore a good tradeoff between the empirical risk and the complexity term needs to be found. Using a technique which was first proposed in [BM92] and later used for SVMs in [CV95], we introduces slack-variables ξ_i to relax the hardmargin constraints

$$y_i(\langle w, \Phi(x_i)\rangle + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, ..., N$$

additionally allowing for some classification errors. The SVM solution can then be found by

1. keeping the upper bound on the VC dimension small
2. minimizing an upper bound $\sum_{i=1}^{N} \xi_i$ on the empirical risk, i.e., the number of training errors.

Thus, we minimize

$$\min_{w,b,\xi}\left\{\frac{1}{2}\|w\|^2 + \mathcal{C}\sum_{i=1}^{N}\xi_i\right\}$$

where the regularization constant $\mathcal{C} > 0$ determines the tradeoff between the empirical error and the complexity term. More $\mathcal{C}$ is big, more the bad classifications are penalized and more the complexity of the class of decision functions is big. This leads to the dual problem:

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

$$\text{subject to } \begin{cases} 0 \leq \alpha_i \leq \mathcal{C}, i = 1, ...N \\ \sum_{i=1}^{N} \alpha_i y_i = 0 \end{cases}$$

From introducing the slack-variables ξ_i, we get the box constraints that limit the size of the Lagrange multipliers: $\alpha_i \leq \mathcal{C}, i = 1, ...N$.

Most optimization methods are based on the second-order optimality conditions, so called Karush-Kuhn-Tucker conditions which state necessary and in some

cases sufficient conditions for a set of variables to be optimal for an optimization problem. It comes handy that these conditions are particularly simple for the dual SVM problem:

$$\begin{array}{lcll} \alpha_i = 0 & \Rightarrow & y_i f(x_i) \geq 1 & and \quad \xi_i = 0 \\ 0 < \alpha_i < \mathcal{C} & \Rightarrow & y_i f(x_i) = 1 & and \quad \xi_i = 0 \\ \alpha_i = \mathcal{C} & \Rightarrow & y_i f(x_i) \leq 1 & and \quad \xi_i \geq 0 \end{array}$$

Therefore, the nonzero coefficients α_i are those associated with the samples x_i of the learning set which are on the margin (i.e. $\alpha_i = \mathcal{C}$ and $y_i f(x_i) < 1$) or inside the margin area (i.e. $0 < \alpha_i < \mathcal{C}$ and $y_i f(x_i) = 1$). Those samples are called *support vectors*.

Thus, the SVM classification is based on a limited set of vectors coming from the learning set which are critical for the classification: the support vectors are the closest data to the decision border (fig. 3.5). Therefore, the outliers which are generally far from the decision border and which introduce a distorted decision are not considered by the SVM classification.

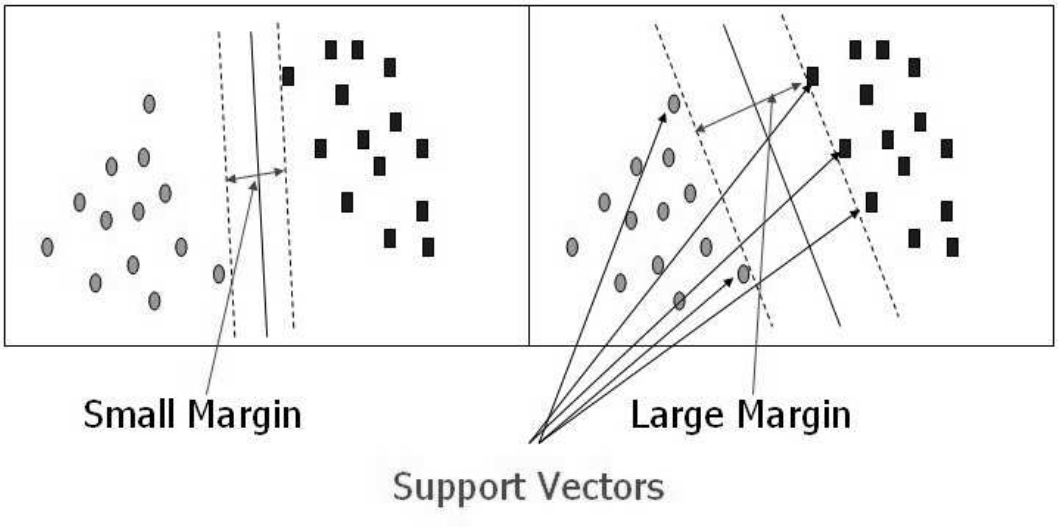

Figure 3.5: Support Vectors

3.3 Conlusion

Support Vector Machines is a tool becoming powerful by the use of kernel functions. It permits the mapping of the data in the feature space so that a separating hyperplane can be found. In our project, we use a java implementation of the Support Vector Machines concept realized by [CL01].

Now that the tool used for the knowledge discovery in our project has been introduced, the way we process information contained in the news coming from the Internet will be explained in the following. Each step is described until the confection of the input file that is sent to the SVM prediction tool.

Chapter 4

Project Description

It is necessary to relocate our work in the whole project of exchange rate prediction using Internet news and DataMining techniques. Indeed, before our arrival at the university of Technology (Sydney), some people had worked on a part of that project and we have had to continue the project using some functionalities already developed.

4.1 Global Project

The global project consists in five big steps:

1. Retrieving of economic-related news from the Internet and organization of those in specific fields (Date, Abstract, Title,...) in order to store them in a database.

2. Manual classification of the stored news depending on their effect on the US Dollar rate.

3. Analysis of the classified news and retrieving of specific information contained in each news belonging to a determined category (the category depends on the effect of the news on the US Dollar rate).

4. Prediction of the categorization of an incoming news using the specific mining information retrieved during the previous step.

5. Using the categorization process, estimation of the variation of the exchange rate between the Euro and the US Dollar.

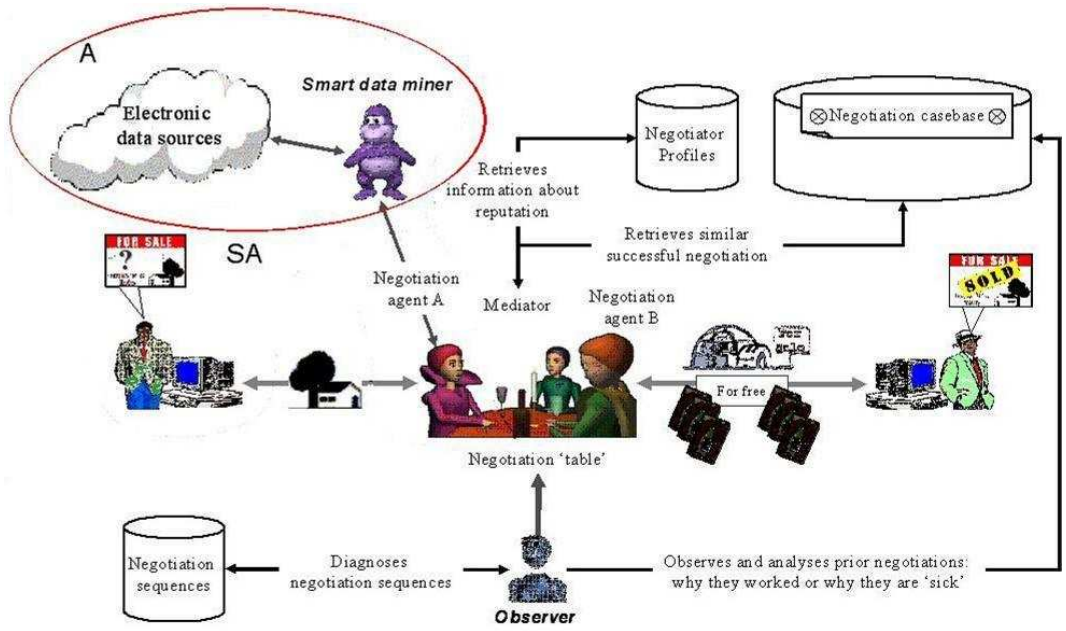

Figure 4.1: Global Project

Those steps are just summarized here but a deepest description will be done afterwards.

The figure 4.1 shows the whole process. The news retrieved are mined by a smart data miner retrieving information for agent A that will use it for getting the best of his negotiation with agent B.

4.2 Overview of our Contribution

Here is a complete description of what was done and what was still to do for each step of the project when we started working on it:

News Retrieval: The news were already retrieved from the internet. We have added a functionality that allows to avoid economic news giving the same information as another news already downloaded. Thus, with that functionality, we have: [`1 information = 1 news`]. It will be described in detail afterwards.

Manual Classification: We have done the manual classification of the news. This step is explained afterwards.

Mining process: The different mining methods were already implemented, but some errors were present. We have corrected bugs and have linked those functionalities together.

Prediction: We have downloaded the prediction tool from the Internet and then we have linked it with the functionalities presented above.

Estimation of the Exchange Rate: This part still has to be done. As shown afterwards, the mining technique has first to be improved before caring about the exchange rate estimation.

4.3 Description of a news

The raw materials of our application are data news which are taken from the Bloomberg web site[1]. Indeed, all our work is based on these news. These data news, once downloaded, are stored in a database. Thenceforth, our application interacts with that database.

The action of introducing news in the database was already done when we started working on the project. Thus, we consider that news located in the database have been correctly introduced (come from Bloomberg web site).

A news is constituted of numerous fields:

1. ***URL*** contains the web site address of this news.
2. ***Date*** indicates the publication date of this news.
3. ***Time*** indicates the publication hour of this news.
4. ***Title*** contains the news headline.
5. ***Summary*** is the summary of the news.
6. ***Source*** indicates the news source (Bloomberg in our case).

[1] www.bloomberg.com

7. ***Article*** contains the text of the news.

Among all these fields, we mainly used three of them: the headline, the summary and the text of the news. Headline is mainly used for the data preparation, namely during operations to remove the redundant news. The summary is necessary for the manual classification. Indeed, the summary is used to deduce the category of the news (related, unrelated, good, bad). Finally, the text of the news is used for Text Mining methods. The field "date" is notwithstanding used for the presentation of the news in the user interface of our program.

4.4 Related work

The methodology we have used is similar on some points to the techniques developed by L. Goffinet and M. Noirhomme-Fraiture in a paper titled "*Automatic cross-referencing of HCI guidelines by statistical methods*". This paper ([GNF99]) suggests some techniques that we could use in our case.

We will now explain the main methods of this technique.

The first method we could take into account is the method that looks after the suppression of insignificant words. After the suppression of those meaningless words, other methods are proposed in order to track down the prospective keywords. For example, a method is used to reduce the size of the words with a view to find the stem of these different words. In the same way, a thesaurus can be used in order to replace the existing synonyms.

The second part of this paper proposes some methods to compute the "weight" given to each keyword. They explain three methods among which we use one. This method is based on the number of occurrences of each keyword inside a document.

By comparing our work and this paper, we notice that some techniques are similar. Indeed, in our work, we have used some techniques in order to eliminate insignificant words and some methods whose the goal is to prepare the Text Mining process. In the same way, the frequency calculation is based on the same idea.

Chapter 5

Core of our work

The implementation part of the project will be analyzed in this chapter. We will first describe the learning process: the refining of the news package downloaded from the internet and how those news are then manually classified in different categories. After this, the mining process of the news will take place and the last step will be the description of the user interfaces we realized for the program. ta Preparation

5.1 Data Preparation

As explained in the chapter 1, databases contain sometimes some irregularities. In our project, it was the case. Our database contained some updated news and different news with the same information but a different headline.

To resolve these two abnormalities, we firstly have implemented two algorithms. In the second part of this section devoted to the Data preparation, we will explain the news classification, one important step of our Text Mining process.

5.1.1 News Refining

As we have explained you, we use data that arise from Internet and especially from a website. That website is "www.bloomberg.com". When news are received, these news are automatically placed in the database. When we have analyzed these different news, we have noticed that some news were redundant, that is,

that the same news was sometimes expressed by several news which give the same information. It exists different redundances:

- updated news.
- news giving the same information, but with a different title.

The first redundance can be called "the updated news". Indeed, we have noted that in the database, some news are just an update of another news. Hereunder, an example (taken from the database) illustrates this redundance.

Dollar May Resume Four-Month Rally Against the Euro
Dollar May Resume Four-Month Rally Against the Euro(Update1)
Dollar May Resume Four-Month Rally Against the Euro(Update2)
Dollar May Resume Four-Month Rally Against the Euro(Update3)
Dollar May Resume Four-Month Rally Against the Euro(Update4)
Dollar May Resume Four-Month Rally Against the Euro(Update5)

Bloomberg, 2005-07-12

We can see in this example that the five last news are useless. That is for this reason that those news will be removed and, only the first one will be conserved. That is a choice to keep the first one, but the other news have the word "update" in their headline. Thus, to facilitate our implementation, we have chosen to keep the first news.

To resolve that redundance, we have reviewed the database to eliminate all "updated news" seeing that the text mining algorithm uses the news number to determinate the exchange rate. Therefore, we implemented a new method.

The second redundance we have discovered specifies that some news contain the same information but with a different title. With the same manner and for the same reason, the database has been sifted with intent to remove superfluous news.

Below, two examples illustrate this redundance. The first one is about the crude oil. The contents of these two news is the same and thus, they give the same

information. Therefore, one of these news will be removed with the algorithm presented farther.

"Crude Oil Futures May Fall as Mild Weather Trims U.S. Heating Oil Demand"

Crude Oil Futures May Fall as Mild Weather Trims U.S. Heating Oil Demand Feb. 7 (Bloomberg) – Crude oil futures may fall, extending last week's 1.5 percent decline, on speculation mild weather will trim demand in the U.S. Northeast, where about 80 percent of the nation's heating oil is consumed. Temperatures in the Northeast may be 8 degrees Fahrenheit.

Bloomberg, 2005-02-07

"Crude Oil Futures Fall as Mild Weather Trims U.S. Demand for Heating Oil"

Crude Oil Futures Fall as Mild Weather Trims U.S. Demand for Heating Oil Feb. 7 (Bloomberg) – Crude oil futures fell, extending last week's 1.5 percent decline, on speculation mild weather will trim demand in the U.S. Northeast, where about 80 percent of the nation's residential heating oil is consumed. Demand for home heating fuel may decline to 26 percent.

Bloomberg, 2005-02-07

The second example sets the same problem with two news with the same idea but with a different headline.

"Group of Seven Says Global Growth Robust, Maintains Stance on Currencies"

The Group of Seven industrial nations pledged to take individual steps to maintain "robust" global growth, while repeating its call for Asian nations to adopt "more flexible" exchange rates.

Bloomberg, 2005-02-07

"Group of Seven Says Global Growth Is Robust, Maintains Its Currency Stance"

The Group of Seven industrial nations pledged to take individual steps to maintain "robust" global growth, while repeating its call for Asian nations to adopt "more flexible" exchange rates.

Bloomberg, 2005-02-07

To resolve this redundance, we have built an algorithm. This algorithm needs three parameters, namely:

1. ***The period of the redundance control***
   This first parameter defines the period on which the algorithm works. Let T, the date that we analyze. If t is the value of this parameter, then we will work on a period of [T, T + t].

2. ***The length of the ignored words***
   This parameter contains the value of the ignored words. For example, if this parameter has a value of 3, then all words whose their length is strictly lower will be ignored. Accordingly, words such as "the", "in", "for", ... will be ignored in our example.

3. ***The threshold***
   The last parameter of this algorithm is a kind of threshold. Indeed, if the number of common words between two news is at least equal to this threshold, then one of these two news will be dropped.

These parameters can be chosen by the programmer. Personally, we had chosen a value of 2 for NbreDays, 6 for MappingNumber and 3 for LengthDroppedWords. With those values, we have had good results. This algorithm is presented hereunder in pseudo code (the implementation of this algorithm in the language Java is available in annexes):

```
Input : A set of news 'setNews'.

Output: A set of news without redundance.
```

```
  const
      nbreDays: the period of the redundance control

      LengthDroppedWords: the length of the ignored words

      nbrMapping: a threshold

  var
      Nbre : integer

Begin:
        For each date of setNews, build a news subset of a determined
        period [Date, Date + nbreDays]

        For each subset, break down the title into words (by removing
        punctuation marks and words whose the length is smaller than
        LengthDroppedWords)

        For each news of this subset, compare the title words with the
        other news titles
            Nbre := number of common words with the other news
            if Nbre >= nbrMapping
                Remove this news

End.
```

Once these steps are realized, the database is ready for the manual classification and later for treatment by the text mining algorithm.

5.1.2 News Classification

Before incorporating the news effect into the exchange rate model, it is necessary to identify the news that are relevant and classify them into "good" or "bad" news categories. This categorization is explained below.

This step is an important step. Indeed, this step implies the suite of the process. A good classification will imply good results, but the inverse is also right, namely,

a bad classification will imply bad results! In order that computer realizes this classification automatically, it is necessary that the computer learns how to do it. That is for this reason that a manual news classification that will learn to the computer how to classify is firstly realized. Moreover, the automatic classification can be realized by using text mining techniques.

From all news we have in the database, we have to classify those one into different categories. The first step of this classification is a news separation into two subclasses, namely: "related news" (news that could have an effect on the economy, and thus that could affect the exchange rate) and "unrelated news" (news that do not have any effect on the economy). To accomplish this separation, it is required to answer a question. This question is: "Could the news have an economic effect ?". If the response is "yes", then this news is declared "related", otherwise this news is qualified of "unrelated".

To answer this question, it exists factors to classify the news. Factors that affect currency exchange rate can be macroeconomic data, statements by central bankers and politicians and political events. For example, here is four news we have to classify as related news or unrelated news:

"France Tries 16 People for Mont Blanc Tunnel Fire That Killed 39 in 1999"

Sixteen people went on trial before a French court today for the 1999 Mont Blanc fire that killed 39 people in the tunnel linking France and Italy under the Alps' highest mountain, a court spokeswoman said.

Bloomberg, 2005-02-07

"Fed Policy Makers Saw Higher Inflation Risks, Minutes of March Meeting Say"

The Federal Reserve's Open Market Committee saw higher inflation risks at its March 22 meeting and debated whether to jettison language about sticking to a "measured" pace of rate increases,records show.

Bloomberg, 2005-04-13

"Dollar May Advance Against Euro on Outlook for Higher U.S. Interest Rates"

The dollar may rise against the euro on expectations a government report tomorrow will

show U.S. wholesale prices rose last month, suggesting the Federal Reserve will continue raising interest rates to subdue inflation.

Bloomberg, 2005-04-18

"Max Schmeling, German Boxer Who Was Joe Louis's Friend and Foe, Dies at 99"

Max Schmeling, the German heavyweight boxing champion who was miscast as a symbol of Adolf Hitler's Nazi regime in his 1938 bout with Joe Louis, died in his native Germany. He was 99.

Bloomberg, 2005-02-07

Among those news, there are two related news and two unrelated news. The first and the fourth news will be classified "unrelated". Contrariwise, the related news are the second and the third news.

After this first classification, we can forget some news (those that are unrelated). For the other news, another classification is realized. This classification is extremely difficult. Indeed, two news can be totally different by using nonetheless a similar set of keywords. For example (taken from [ZSD05]), here are two news with nearly the same words. Nonetheless, those two news have an opposite effect on the rate of the Dollar.

1. The interest rate has gone up. The US dollar has gone down
2. The interest rate has gone down. The US dollar has gone up

In this example, the two news use the same set of keywords, that implies some difficulties for the classification. That is for this reason that text mining techniques used for the automatic classification are very important and have to be effective.

[ZSD05] informs us that recent studies points out that the effect of the news is the combined effect of market expectations and the news itself. Nevertheless, in our work, we only took into account the news itself and not the market expectations. The market expectations will be incorporated into the model in a later stage but not during our work.

This second classification (realized on the "related news") is again the result of a question, namely: "Could the news have an effect on the US Dollar rate?".

In this second classification, there are four different types whose the definitions are:

Good : a related news is classified as "good" when the news has a positive effect on the US dollar currency.

Bad : a related news is classified as "bad" when the news has a negative effect on the US dollar currency.

No Effect : a related news is classified as "no_effect" when the effect of the news is not important enough to be considered.

Other : a related news is classified as "other" when the news does not have an effect neither on the US dollar currency neither on the Euro. For example, when the subject of a news is the Japanese Yen.

If we take the example above (from [ZSD05]), the first news will be classified as a "bad" news and the second one as a "good" news.

To realize this second classification, it exists some factors to determine if a news will be classified as "good" or "bad". In the table 6.7, we can see the main factors and their effects on the US dollar currency.

Factor	Factor effect	good/bad
interest rate goes up in US	US dollar goes up	good
interest rate goes down in US	US dollar goes down	bad
unemployment rate goes up in US	US dollar goes down	bad
unemployment rate goes down in US	US dollar goes up	good
gross national product (GNP) goes up in US	US dollar goes up	good
gross national product (GNP) goes down in US	US dollar goes down	bad
gross domestic product (GDP) goes up in US	US dollar goes up	good
gross domestic product (GDP) goes down in US	US dollar goes down	bad
inflation goes up in US	US dollar goes down	bad
inflation goes down in US	US dollar goes up	good

Table 5.1: Classification factors and their effects

Let us note that a "good" news for the US dollar corresponds to a "bad" news for the Euro and that a "bad" news for the US dollar corresponds to a "good" news for the Euro. The table with the main factors and their effects on the Euro is available in annexes.

To illustrate those concepts, here is some examples of "good", "bad", "no effect" and "other" news. Let us begin with two "good" news. The first one is a "bad" news for the Euro, so a "good" news for the US dollar. The second one is a "good" news for the US dollar and thus a "bad" news for the Euro.

"Industrial Production Fell in February as Record Oil Prices Raised Costs"

Industrial production in Germany, Europe's largest economy, fell the most in more than two years in February as record oil prices increased costs for companies and eroded consumers' purchasing power, denting growth prospects.

Bloomberg, 2005-04-08

"Dollar May Advance Against Euro on Outlook for Higher U.S. Interest Rates"

The dollar may rise against the euro on expectations a government report tomorrow will show U.S. wholesale prices rose last month, suggesting the Federal Reserve will continue raising interest rates to subdue inflation.

Bloomberg, 2005-04-08

Next, two "bad" news for the US dollar. Again, the first one is indirectly a "bad" news since this news is "good" for the Euro. The second one is a "bad" news directly linked with the US dollar.

Business Confidence in Germany, France, Italy Probably Gained in February

Business confidence in Germany, France and Italy, the three largest economies in the 12-nation euro area, probably increased this month amid signs growth may rebound from a slowdown in the fourth quarter, surveys of economists showed.

Bloomberg, 2005-02-21

U.S. Consumer Prices Increase 0.1 Percent, Easing Concerns About Inflation

Prices paid by U.S. consumers rose 0.1 percent in January, easing concern that surging commodities costs would spur faster inflation.

Bloomberg, 2005-02-24

Here are now two news to illustrate news that have "no effect". We can see that those news are economic news but that their effects are not important enough to be considered.

"Group of Seven Says Global Growth Robust, Maintains Stance on Currencies"

The Group of Seven industrial nations pledged to take individual steps to maintain "robust" global growth, while repeating its call for Asian nations to adopt "more flexible" exchange rates.

Bloomberg, 2005-02-07

"EU Fails to Agree on Proposed Relaxing of Budget-Deficit Borrowing Limits"

European Union finance ministers failed to agree on a proposed relaxation of budget-deficit rules as Germany demanded greater scope to increase spending or cut taxes to spur the faltering economy.

Bloomberg, 2005-03-09

Finally, two examples for the "other" news. As you can see, those news involve the Australian Central Bank and the Japanese Yen. These news have an economic effect but not directly on the US dollar and on the Euro.

"Australian Central Bank Says Rate Increase More Likely on Inflation Signs"

Australia's central bank said the likelihood of an interest-rate increase "in the months ahead" has grown after the inflation rate doubled in the fourth quarter.

Bloomberg, 2005-02-07

"Yen Rises After Japan's Industrial Production Expands More Than Forecast"

The yen rose in Asia after a Japanese government report showed industrial production expanded more than expected, helping the world's second-largest economy recover from recession..

Bloomberg, 2005-02-28

To resume these classifications, the schema on the figure 5.1 illustrates the two classifications: classification as "related" or "unrelated" news, and for "related" news, classification as "good", "bad", "no effect" or "other" news.

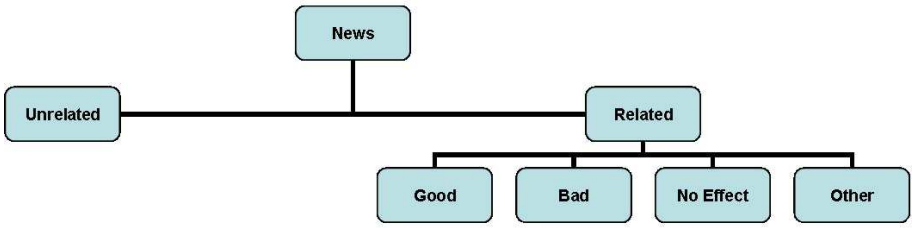

Figure 5.1: Summary of the news classification

5.2 Text Mining Process

Now that the news have been manually classified, we will use that classification to predict how an incoming news should be classified by the computer. The following steps explain how the computer will learn from the manual classification and thus how he will choose a category of classification for an incoming news.

5.2.1 Keywords Extraction

Assume the computer has to be able to determine if a news is a good or a bad news for the dollar. Therefore, we will have to train the computer with two packages of news: a package of "good" news and a package of "bad" news. Those two packages will separately follow the process of keywords extraction.

The former DataMining processes were based on the nature of the word. For example, the word "`house`" was classified as a name and the word "`accepted`" was classified as a verb. It was a big syntactic analysis.

The technique we use here is different. In fact, a syntactic analysis is too time-consuming on a big range of data. Therefore, the new mining techniques have a different way of processing. The syntactic analysis is no longer used. The mining is based on the occurrence and co-occurrence of terms.

The keywords extraction process is divided in three steps: news cutting, stop-word method, stemmer method. For explaining these, we will use the following news as an example:

"Fed's Bies Says Inflation 'Under Control,' Rates Will Continue to Increase"

Fed's Bies Says Inflation 'Under Control,' Rates Will Continue to Increase Listen Feb. 7 (Bloomberg) – The U.S. economy is strong and inflation is under control so the Federal Reserve will continue to raise rates at a "measured" pace, Governor Susan Bies said. Inflation, as measured by the personal consumption expenditures index excluding food and energy, is between 1.5 percent and 2.5 percent a year, Bies said. "On average prices are well-contained," she said in a speech at the University of Tennessee in Martin. "We're committed to a measured pace of continuing to raise interest rates." Bies's comments mirrored those made by the Fed's policymaking Open Market Committee after last week's meeting. The FOMC raised the benchmark U.S. interest rate a quarter point to 2.5 percent, and said it would continue to raise rates at a measured pace. The committee also said inflation is "well contained." The rise in oil prices in the last year "got the Fed's attention," Bies said because it took money out of consumer pocketbooks that would have been spent on other goods and boosted the U.S. trade deficit, contributing to the decline in the dollar. Bies said that with the large trade and current account deficit, the dollar would fall further if it were not being boosted by foreign governments. "Our trading partners intervene aggressively," she said. "The Chinese are pegging their currency. This is not a free market." Bies answered questions after a speech at University of Tennessee in Martin. She said the U.S. still attracts "a tremendous amount of foreign direct investment" that helps finance the trade and U.S. fiscal deficit. The federal budget deficit, which helped stimulate the economy when it was ailing, also needs to be brought under control, she said. "The economy now is growing at a healthy pace and you don't need that extra stimulus," she said. Bies said President George W. Bush's proposal to divert some payroll taxes from social security into private accounts is a "different issue" from making the retiree program solvent. She said most people are not educated enough to responsibly invest in private accounts. "You can't just offer it. You've got to really dedicate a lot to consumer education," she said.

Bloomberg, 2005-02-09

Despite the Mining process will analyze thousands of news, we will follow the way of only one of these for an easier understanding.

News Cutting

The first step is the cutting of the sentences (contained in the news) in a table of words. First, a preliminary point is needed: the removing of all digits contained in the news and all punctuation characters such as , ? ; : ' ! etc. Then, the sentences cutting can be done. For a question of clarity, the tables containing all the sentences of the news will be placed in appendixes, so that the tables shown here will contain only one sentence of the news. The sentence used is:

Inflation, as measured by the personal consumption expenditures index excluding food and energy, is between 1.5 percent and 2.5 percent a year, Bies said.

After the news cutting, the table 5.2 is thus obtained (for the table containing all the sentences, see appendix 2, table 6.4).

Inflation	as	measured	by	the
personal	consumption	expenditures	index	excluding
food	and	energy	is	between
percent	and	percent	a	year
Bies	said			

Table 5.2: News Cutting

StopWord Method

Now that all sentences of the considered news have been cut, we can apply the StopWord method. This method check all the words contained in the previous table and, if one of them is contained in the StopWord list, this word is deleted from the table. The StopWord list contains the most common words (such as those in the table 5.3). Those words are not important for the learning process of the computer because they do not give any interesting information. For example, the words "by" and "they" can be found in lots of sentences, independently of the

a	about	above	according	across	actually
adj	after	afterwards	again	against	all
almost	alone	along	already	also	although
always	among	amongst	an	and	another
any	anyhow	anyone	anything	anywhere	are
aren't	around	as	at	be	became
because	become	beyond	has	had	how
everything	except	even	ever	here	he
ie	if	its	itself	last	later
made	many	me	maybe	like	mrs

Table 5.3: StopWords list

meaning of the sentence.

This method was already implemented when we started working on the project, but we corrected the following bug: Before, the sentence:

```
"He says:  It's a beautiful day"
```

was transformed in

```
"StopWord says StopWord's StopWord beautiful day"
```

and so, words kept were: `says, StopWord's, beautifull day`. We corrected the bug so that the word `StopWord's` is not kept anymore as a word in the list of words.

After the application of this method, all the StopWords are deleted from the list of words and we obtain the table 5.4 [1].

Stemmer Method

Now that the most important words contained in the sentences of the news have been retrieved, a last step is needed before the calculation of the frequency of the

[1] for the table containing all the sentences, see appendix 2, table 6.5

Inflation	measured	personal	consumption	expenditures
index	excluding	food	energy	percent
percent	year	Bies	said	

Table 5.4: StopWord Method

words. This method is called "Stemmer". The first thing to be done is the removal of every uppercase for a lowercase. According to [PCC03] "Stemming means finding and returning the root form (or base form) of a word. Stemming enables the investigator to work with linguistic forms that are more abstract than those of the original text. For example, the stem of `grind`, `grinds`, `grinding`, and `ground` is `grind`. The document collection often contains terms that do have the same base form but share the same meaning in context. For example, the words `teach`, `instruct`, `educate` and `train` do not have a common stem, but share the same meaning of `teach`. Text mining can relate words with similar stems. The capability can be extended to numeric codes".

The stemming method used here is not as complete and complex as the one described above. The meaning of the words is not taken into account. For example, the words `look` and `watch`, despite their identical meaning, will not be converted into one single word. The method will only remove surface markings from words to reveal their basic form. For example,

- `forms` → `form`
- `forming` → `form`
- `formed` → `form`
- `former` → `form`

The table 5.5 represents a part of the end-of-words conversions used in the project. Once the stemmer method has been applied to the table 5.4 we obtain the table 5.6 [2].

[2]for the table containing all the sentences, see appendix 2, table 6.6

..ational	→	..ate	..tional	→	..tion
..enci	→	..ence	..anci	→	..ance
..izer	→	..ize	..bli	→	..ble
..alli	→	..al	..entli	→	..ent
..eli	→	..e	..ization	→	..ize
..fullness	→	..ful	..ousness	→	..ous
..aliti	→	..al	..biliti	→	..ble
..icate	→	..ic	..alize	→	..al
..iciti	→	..ic	..ical	→	..ic
..logi	→	..log	..alism	→	..al

Table 5.5: Stemming Table

inflat	measur	person	consumpt	expenditur
index	exclud	food	energi	percent
percent	year	bi	said	

Table 5.6: Stemmer Method

5.2.2 Frequencies Calculation

Now that every news of the two packages ("good" and "bad" news) went through the keywords extraction process, the following steps will take place:

Inter-Package Pooling

For every package ("good" and "bad" news or "related" and "unrelated" news) a pooling of all the keywords extracted is done. Consequently, we obtain two lists of keywords, one for each package. For instance, let us assume the tables 5.7 and 5.8, the table 5.9 represent the pooling of the package containing two news: A and B.

contain	rise	oil	price	year	got
larg	trade	current	account	deficit	dollar
answer	question	speech	univers	rise	martin

Table 5.7: Inter-Package Pooling (1)

realli	dedic	rise	consum	educ	said
benchmark	interest	rate	quarter	point	dollar

Table 5.8: Inter-Package Pooling (2)

contain	rise	oil	price	year	got
larg	trade	current	account	deficit	dollar
answer	question	speech	univers	martin	realli
dedic	consum	educ	said	benchmark	interest
rate	quarter	point			

Table 5.9: Inter-Package Pooling (3)

Keywords Frequency Calculation

For each of the two lists of keywords obtained by the pooling, the frequency of each keyword will be calculated. For example, assume the list in the table 5.9. For each word of this list, we will sum the number of times this word appears in the news of the considered package. The table 5.10 will thus be obtained.

contain	1	rise	3	oil	1
price	1	year	1	got	1
larg	1	trade	1	current	1
account	1	deficit	1	dollar	2
answer	1	question	1	speech	1
univers	1	martin	1	realli	1
dedic	1	consum	1	educ	1
said	1	benchmark	1	interest	1
rate	1	quarter	1	point	1

Table 5.10: Keywords Frequency in one package

5.2.3 Keywords Selection

Once the two frequencies tables have been created, the following step is to add in each of these two tables a new column containing the frequency of that word in the other package of news. For example, assume two tables 5.11 and 5.12 contain-

ing respectively the list keywords-frequencies of the news package A and the news package B. Those two tables will be thus transformed into the tables 5.13 and 5.14 respectively.

dollar	120	advanc	9	month	25
high	12	greenspan	29	eas	1
deficit	43	worri	2	feb	38
bloomberg	21	trade	49	euro	51
feder	12	reserv	15	chairman	9
alan	6	damp	2	concern	7
widen	4	current	17	account	18
spur	3	unpreced	1	fourth	3
straight	2	said	3	drop	102

Table 5.11: Frequencies of the keywords belonging to the package A

european	12	manufactur	6	growth	23
probabl	9	pick	7	survei	11
show	9	feb	9	bloomberg	9
dozen	2	countri	6	share	2
euro	16	accel	5	second	2
month	12	januari	10	retreat	2
dollar	110	month	13	account	23
said	7	concern	10	eas	1
record	8	sign	4	domest	2

Table 5.12: Frequencies of the keywords belonging to the package B

The keywords on these tables (5.13 and 5.14) are organized in a certain order. In fact, the first keywords (left to right and up to down) are the most "interesting" for the training of the computer because those words have the most "favorable" gap between their total of frequencies in the news belonging to the package A and in the news belonging to the package B. For example, the word `drop` appears 102 times in the news contained in the package A and 0 times in the news contained in the package B. Thus, if this word was found in a news, it could be an interesting

drop	102	0	deficit	43	0	trade	49	0
greenspan	29	0	current	17	0	reserv	15	0
feder	12	0	high	12	0	chairman	9	0
advanc	9	0	alan	6	0	widen	4	0
fourth	3	0	spur	3	0	straight	2	0
damp	2	0	worri	2	0	unpreced	1	0
eas	1	1	said	3	7	concern	7	10
feb	38	9	bloomberg	21	9	month	25	12
euro	51	16	account	18	23	dollar	120	110

Table 5.13: frequencies of the keywords belonging to the package A in the two packages

growth	0	23	oil	0	13	european	0	12
survei	0	11	januari	0	10	probabl	0	9
show	0	9	record	0	8	pick	0	7
manufactur	0	6	countri	0	6	accel	0	5
sign	0	4	domest	0	2	second	0	2
dozen	0	2	share	0	2	retreat	0	2
eas	1	1	said	3	7	concern	7	10
feb	38	9	bloomberg	21	9	month	25	12
euro	51	16	account	18	23	dollar	120	110

Table 5.14: frequencies of the keywords belonging to the package B in the two packages

criteria to determine if that news belong to the package A or to the package B.

Depending on the power of the machine on which the program runs, the programmer has to choose the number (X) of keywords to be considered for the analysis. The first X/2 keywords of each package will be retained. For example, let us consider the table 5.13 containing the keywords of the package A and the table 5.14 containing the keywords of the package B. If, for example, the number (X) of the keywords to be considered for the analysis was equal to 18, the final table of the keywords extraction process would be the table 5.15 containing 9 keywords from the package A and 9 keywords from the package B.

drop	102	0	growth	0	23	deficit	43	0
oil	0	13	trade	49	0	european	0	12
greenspan	29	0	survei	0	11	current	17	0
januari	0	10	reserv	15	0	probabl	0	9
feder	12	0	show	0	9	high	12	0
record	0	8	chairman	9	0	pick	0	7

Table 5.15: Frequencies of chosen keywords in the two packages (A and B)

5.2.4 Formatting

Now that the keywords selection has been done, we can care about their frequencies in each considered news. For example, assume a package A containing 8 news and a package B containing 3 news. For each of those news, we will count the number of times each selected keyword is encountered. All that information will then be sent to the learning process of the computer. Depending on the learning process, a specific information format is needed. In our case, we use LIBSVM [CL01] which requires the input format shown on table 5.16.

```
-1 1:4 3:1 5:2 7:3 9:1 11:1 13:1 15:1 17:1 19:1 21:1
-1 1:27 21:3 23:9 25:1 27:2 29:3 31:1 33:4 35:2 37:6 39:1 77:1 79:1 89:1 91:1 93:1 95:1
-1 1:18 3:1 5:6 7:7 9:2 57:2 59:1 61:1 63:3 65:3 69:2 75:1 87:1 91:2 93:1 95:1 97:1 99:1
-1 1:16 9:1 11:1 13:1 15:7 19:3 23:1 37:1 41:1 53:3 57:1 69:1 75:6 81:1 83:2 87:1 89:2 97:2
-1 1:5 3:1 5:3 7:6 9:1 11:1 13:1 17:147:1 53:1 73:2 77:1 83:1 85:1 87:1 89:1 93:1 95:1
-1 41:1 67:1 79:4 97:1
-1 1:17 5:4 7:6 9:1 11:1 15:6 17:1 19:2 21:3 23:9 91:3 93:1 95:1 97:1 99:3
-1 1:3 5:1 7:1 9:1 11:1 15:1 19:1 21:1 23:3 29:1 31:1 33:1 35:2 37:2 99:3
+1 2:2 4:1 6:4 8:1 10:2 12:1 14:1 16:2 18:2 20:1 22:1 24:1 26:2 28:1 30:1 32:1
+1 2:2 6:2 8:1 14:1 26:2 28:1 30:1 38:1 44:2 46:1 48:1 50:1 52:1 54:3 92:1 94:1 96:1 100:1
+1 2:3 4:1 6:3 8:2 10:1 12:1 14:1 16:2 92:1 94:1 96:1 98:1 100:1
```

Table 5.16: SVM input format

In that table, each line represents one news. If the line starts with `-1`, it means this is a news from the package A. The lines containing one news coming from the package B start with `+1`. After the `-1` or the `+1`, the line contains a list of

couples `numberA:numberB`. The `numberA` represent the number of a keyword and the `numberB` is the number of times that keyword has been encountered in the considered news. For example, the couple `31:7` means that the keyword 31 has been encountered 7 times in the considered news.

The SVM prediction algorithm will then use this input to learn how to classify an incoming news. The SVM prediction algorithm is defined in the chapter 3 and the results obtained are presented in the chapter 6.

5.3 User Interface

Interfaces enable users to interact with the system. Our application contains also some interfaces that will be presented in this section. Furthermore, the main characteristics of our interface will also be exposed. Ultimately, we will end with an interfaces critic.

The system to which we have contributed is a multi-agent system. As we can see on the figure 5.2, there are several modules. On this diagram, we can distinguish the different modules that compose the system, namely Text Mining Agent, Exchange Rate Prediction Model and User Interface. Data taken from Bloomberg site is stored in the database. Then, the three modules can accede to the database to get data news. The "Text Mining Agent" module is used to classify news. From that classification, the "Exchange Rate Prediction Model" calculates the exchange rate. The last module, the "User Interface" module is the accessing point to the system.

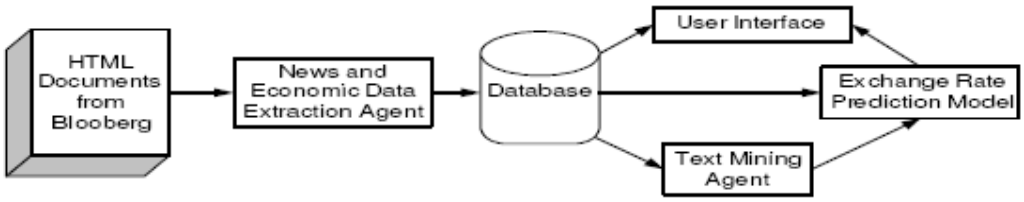

Figure 5.2: The structure of the multi-agent system (taken from [ZSD05])

The objective of this section is to present the main user interfaces. It will be also interesting to analyze the characteristics of the user (and his environment) of our interfaces. Some screen captures will be used to show the main interfaces.

Notice that we did not develop the main interface of the program. The secondary interfaces we added have been realized in accordance with the main one. Thus, we did not use any specific method to develop them. They just follow the way the main interface is implemented.

5.3.1 Interface functionalities

Our application is composed of a main interface and several major menu items. In the main user interface (that we can see on the figure 5.3), the displayed information is the exchange rate data and the news articles. About the news articles, it is possible to visualize the news and the summary for the selected day if the user selects a news. For the exchange rate data, it is possible to display the prediction of the exchange rate and a graphic with the exchange rate evolution. With this interface, the user knows the main information.

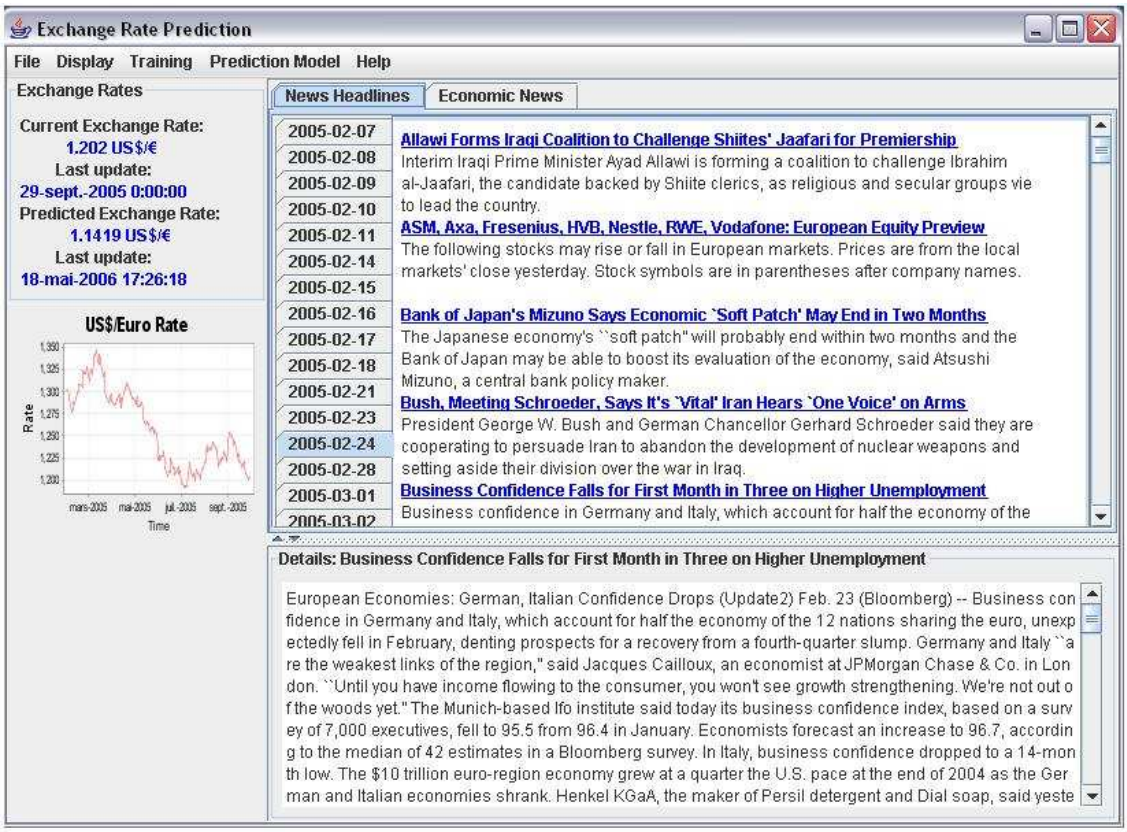

Figure 5.3: The main user interface

To explain the interfaces, we use the process that is used in the normal case, namely: first, the manual classification; then, the automatic classification that we call also prediction and ultimately, the results display.

"Training" menu is essentially used for the training. This menu allows the user

to realize a manual classification as well as to launch a classification on another period that the training period. The user interface realized for the manual classification is displayed on the figure 5.4.

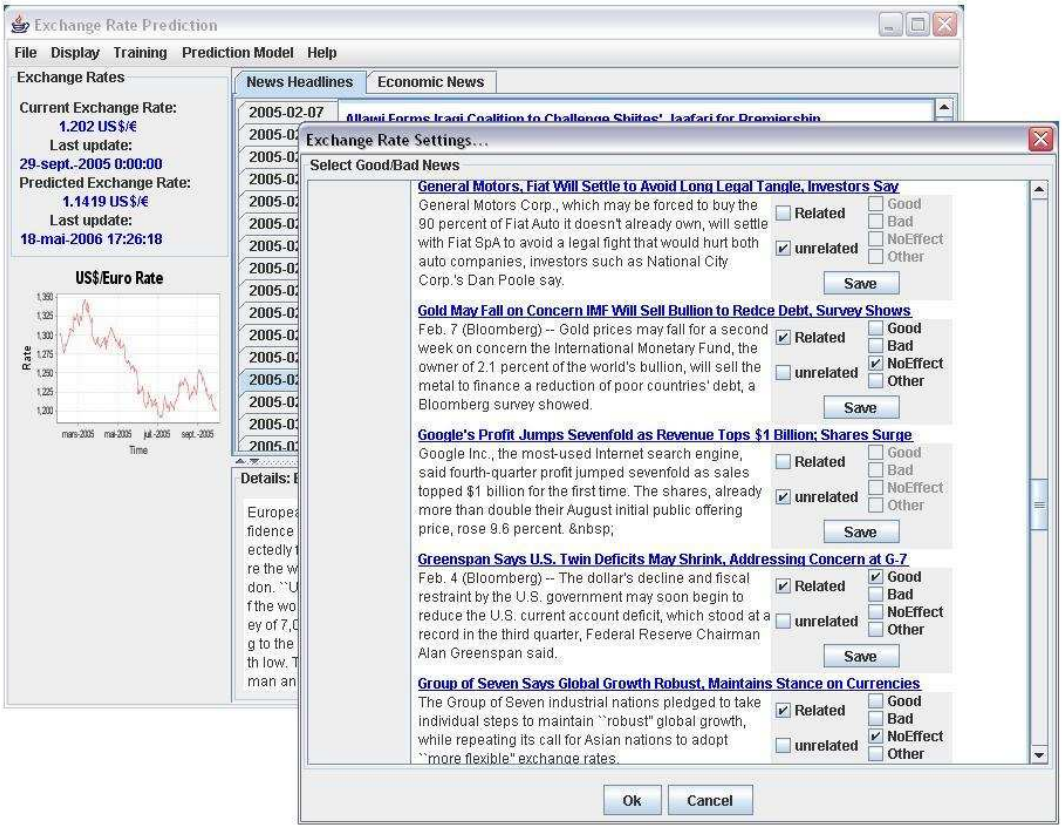

Figure 5.4: The manual classification interface

Once the manual classification is done, the prediction of the automatic classification can be executed. To execute this action, we use the second menu:"Prediction". Available actions in the "Prediction" menu allow to realize a prediction of news classification. After the period selection, the prediction is realized with the results of the manual classification. Once the prediction is done, it is possible to view the results.

Ultimately, the last menu is destined to display classification results, prediction results, manual classification results and classified news. The training results and

prediction results are simply the results of an automatic training or an automatic prediction. The manual classification results display the results of the comparison between the news effect with the exchange rate fluctuation. About the classified news data, it displays news that have been classified (either with a manual classification or with a automatic classification). The user interface for the classified news data can be seen on the figure 5.5.

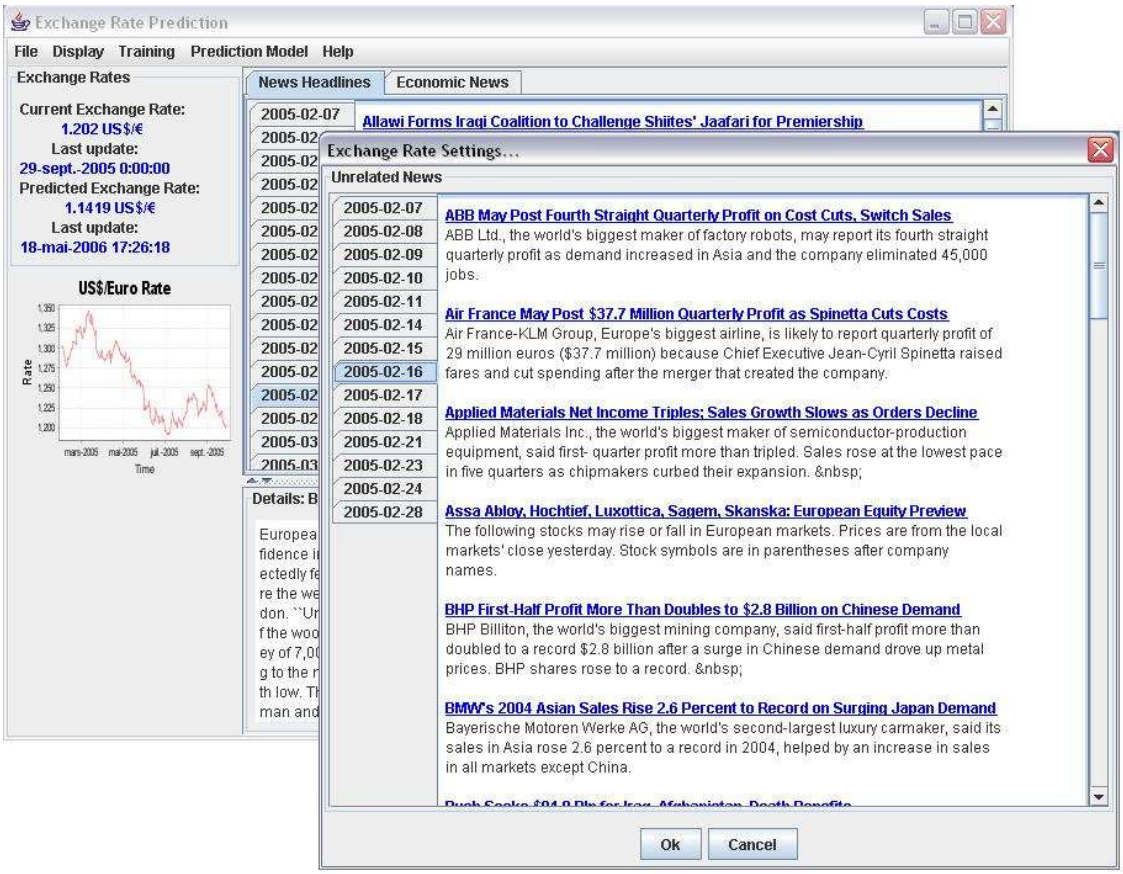

Figure 5.5: The menu of the classified news data

5.3.2 Interfaces critic

Once the main interfaces have been explained, it would be useful to confront the interface elements with the user and his environment. Indeed, our interfaces must be usable by the users. Thus, it is important that the interfaces respect some criteria. We will not critic the ergonomic aspect of the user interfaces, but we will

justify our choices and describe the users and their environment. This analysis is based on the identification check-list of the use context presented in [NF].

Concerning the justification of our choices, it is useful to remark that the layout of the menus has been realized in terms of rapidity and logic. Indeed, the menus are organized in accordance with the typical execution of the application. In the application and in our interface, we begin with the "training" operations. Then, it is the "prediction" actions for the news classification. Ultimately, with the "display" menu, it is possible to visualize the results of the actions realized with the two other menus.

Concerning the "typical user", here are the abilities he should have:

1. have a strong background in economics. Indeed, if the user executes this application because he wants to manually classify the news, he will need a good knowledge in the financial domain.

2. have some experience with computers.

3. be able to understand English. Indeed, English is the language of the application.

The user environment is characterized by an application executed on a personal computer or in a company. There is a constraint for the hardware. Indeed, this application is a text-mining application and asks a lot of resources, especially for the Random Access Memory. To help the users, a documentation guide should be written but the project is not ended yet.

Chapter 6

Personal Contribution

This chapter summarizes the different points of the project we covered during our traineeship at the University of Technology, Sydney. The last section is devoted to the results we obtained with the Support Vector Machine prediction algorithm.

6.1 Achievements

As those achievements have already been presented in this report, we will just summarize them here.

Data Preparation

We first modified the data set necessary for the mining process. As already explained, some news were redundant. Thus, we implemented a method for cleaning the data set.

Manual Classification

The manual classification of news has been time-consuming. Indeed, we classified a big amount of news: 2589 news. Those news are the news released between 2005-02-07 and the 2005-07-05 on the web site of Bloomberg [1]. The table 6.1 contains the different categories in which we classified the 2589 news.

This manual classification was necessary for training the computer to classify news in order to automate this classification.

[1] http://www.bloomberg.com

Related:	704 (27,2 %)	Unrelated:	1885 (72,8 %)
Good:	200 (28,4 %)		
Bad:	113 (16,1 %)		
No_Effect:	230 (32,7 %)		
Other:	161 (22,9 %)		

Table 6.1: Classification Categories

Correction of implemented functionalities

After having linked the different functionalities that were already developed, we remarked that the results we obtained with our first test of the prediction algorithm were suspicious. Thus, we checked all functionalities and discovered some mistakes we corrected:

1. The frequencies calculated while retrieving words from the different news were not always correct. Thus, we modified the frequency calculation algorithm.

2. The keywords selection method has been modified so that the keywords selected are real keywords (See chapter 5 section 2 for more explanations).

3. The method designed to transform the keywords and their frequencies into a correct input format for the Support Vector Machine algorithm has been modified. Indeed, the information given to the SVM algorithm was not complete.

User Interfaces

Even if this is not the main part of our project, the development of user interfaces was time-consuming. We can divide the work concerning the user interfaces into two parts:

1. Improvement of the main interface (already developed) with, inter alia, a new organization of the main menu.

2. Adding of new interfaces and menu linked to the new functionalities of the program that we have positioned.

6.2 Encountered Difficulties

We encountered some difficulties during our traineeship. The first difficulty was the language. It was sometimes difficult to understand the expectations of our supervisor. We also faced an other kind of difficulties, here are two of them:

1. The UTS server containing our database was frequently down. Thus, we have had to repatriate the data on our personal computer and install a local MySQL Server.

2. When we started working on the project, we have had to reuse the java code of the already implemented parts that had been developed by two different persons. Some of the existing programming techniques were difficult to understand.

6.3 Prediction Results

After correction of the bugs described earlier, we have used the news we manually classified to test the prediction abilities of the SVM algorithm.

We did two different tests. The first test was to evaluate the ability of the computer to determine if a news is "Related" or "Unrelated". Therefore, we have used an amount of manually classified news. This first test has been realized on different periods of time. These periods were one week, two weeks, one month and two months. Thus, we have respectively worked on 183, 336, 414 and 962 news. With these news, we have also worked on different percentages for the "training" variable. In our tests, this variable has taken the value of 60, 70 and 80 %. That is the same reasoning for the "test" variable. This variable had for value: 40, 30 and 20 %. For reminding, the "training" variable is the percentage of news that were used in order to train the computer to recognize a related or unrelated news and the "test" variable was used to test if the computer had classified the news in the same category as the manual classification. For example, if a news was manually classified as "related" and, after training, the computer classifies that news as "unrelated", the computer failed his prediction.

The results of these different tests can be seen on the table 6.2.

Number of news	Training percentage	Test Percentage	Percentage of correct classif.
183	60% (111 news)	40% (72 news)	79,2%
183	70% (129 news)	30% (54 news)	77,78%
183	80% (148 news)	20% (35 news)	80%
336	60% (202 news)	40% (134 news)	85,07%
336	70% (237 news)	30% (99 news)	85,85%
336	80% (270 news)	20% (66 news)	87,87%
414	60% (250 news)	40% (164 news)	84,76%
414	70% (290 news)	30% (124 news)	85,48%
414	80% (332 news)	20% (82 news)	86,58%
962	60% (579 news)	40% (383 news)	88,51%
962	70% (675 news)	30% (287 news)	90,24%
962	80% (771 news)	20% (191 news)	92,67%

Table 6.2: Related-Unrelated Classification Test

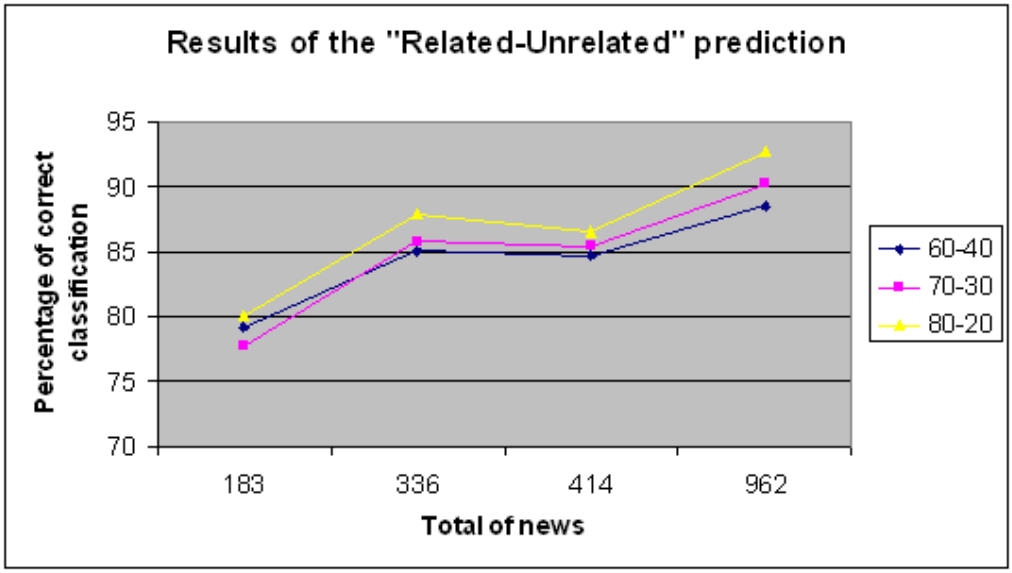

Figure 6.1: Results of the "Related-Unrelated" prediction

These results can be represented by a graph. This one (figure 6.1) shows the results of the "Related-Unrelated" prediction (based on the results of the table 6.2). On the X-axis, there is the number of news on which we have worked. On

the Y-axis, we have the percentage of correct classification.

For the second test, we have used the news that were manually classified as "Related". The goal here was for the computer to classify a related news as a "good" or a "bad" news. The percentages of "training" and "test" are the same as those used for the first test.

The results of that test can be seen on the table 6.3.

Total of news	Training percentage	Test Percentage	Percentage of correct classif.
22	60% (14 news)	40% (8 news)	62,5%
22	70% (17 news)	30% (5 news)	60%
22	80% (19 news)	20% (3 news)	66,66%
42	60% (26 news)	40% (16 news)	68,75%
42	70% (30 news)	30% (12 news)	66,66%
42	80% (36 news)	20% (8 news)	62,5%
54	60% (33 news)	40% (21 news)	57,14%
54	70% (39 news)	30% (15 news)	46,66%
54	80% (44 news)	20% (10 news)	60%
108	60% (66 news)	40% (42 news)	57,14%
108	70% (76 news)	30% (32 news)	56,25%
108	80% (88 news)	20% (20 news)	70%
246	60% (149 news)	40% (97 news)	60,82%
246	70% (174 news)	30% (72 news)	63,88%
246	80% (198 news)	20% (48 news)	70,83%
303	60% (183 news)	40% (120 news)	64,66%
303	70% (213 news)	30% (90 news)	65,55%
303	80% (243 news)	20% (60 news)	68,33%

Table 6.3: Good-Bad Classification Test

The figure 6.2 shows the results of the "Good-Bad" prediction (based on the results of the table 6.3). On the X-axis, there is the number of news on which we have worked. On the Y-axis, we have the percentage of correct classification.

With this graph, it is easy to observe that the results of the second test are less

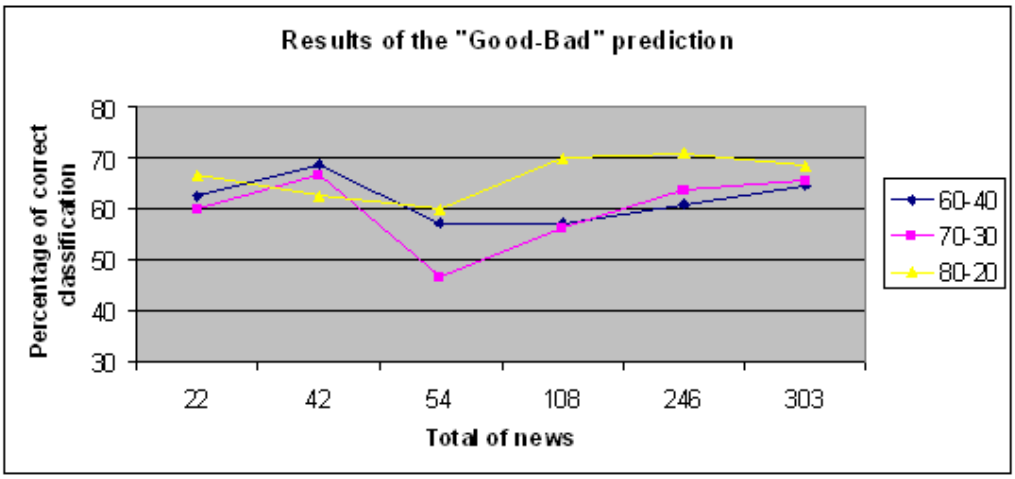

Figure 6.2: Results of the "Good-Bad" prediction

interesting that those from the first test. It is obviously due to the fact it is easier to determine if a news is economy-related than to determine if a news could have a positive or a negative impact on the dollar currency. But we think that those results can also be explain by the fact we are not economists and thus, that our manual classification of news as "good" or "bad" news is not entirely reliable.

Those results have been presented during the Australian Conference on Artificial Intelligence 2005 at the University of Technology of Sydney, Australia.

Conclusion

The objective of this work was to estimate on short term the exchange rate between the US dollar and the Euro. Some points still have to be discussed.

Is the actual mining algorithm powerful enough for this task ? For reminding, the category estimation of a news by the computer goes through two steps. First, the computer has to estimate if the news is "related" or "unrelated". Then, if the news is "related", the computer will estimate if that news is "good", "bad", "no_effect" or "other". If we take a look on the test results, to predict if a news is "related" or not, with a data set of 336 news (with 70% of training and 30% of test), we have 85,85 % of success. Then, with a data set of 42 news classified as "related", we have 66,66% of success while classifying them as "good" or "bad" news. There are two limits with that classification. First, once a news is classified has a "related" news, the next classification is "good" or "bad". Thus, the system does not take into account a news that could be a "No_effect" or "other" news. This is thus the first thing that has to be improved in a future work. Second, if we assume that the news classified as "related" could only be "good" or "bad" news, the total success of the operation is:

$$85,85\% * 66,66\% = 57,22\%$$

Thus, for classifying a news as "good" or "bad" news, we have a total success of 57,22 %. This means that there is about one news on two which would be well classified. We estimate that the results are not satisfactory, this lead us to the conclusion that the algorithm has to be improved. Perhaps we could use another algorithm as the one described in [ZSD05].

However, the bad results could also be explained by the fact that we personally did the manual classification of the news. Since we estimate that our personal

background in economics is not sufficient enough for being sure of the quality of our manual classification. Therefore, we think it could be interesting that the manual classification be done by an economist. After this, if a new test is realized, the obtained results should be more interesting.

With the goal of ameliorate the results, we propose several things for a future work:

1. Improvement of the actual mining algorithm
2. Defining a kind of "importance" for the different news. For example, if a news says `"The US Dollar gains 3% against the Euro"` and another news says `"The unemployment rate could decline"`, the importance of the first news is clearly upper to the second one. That evaluation of the importance of a news would help for the future estimation of the exchange rate evolution.
3. Getting news from different economic-related websites because the actual news are only taken from the Bloomberg website. The fact a news can be found on several economic websites is a gage of legitimacy.

We would like to say that we still think to estimate on short term the evolution of the exchange rate between the US Dollar and the Euro is very complex and perhaps impossible due to the number of different elements to take into account.

To conclude, Text Mining and Data Mining technologies can bring solutions that allow to resolve major business problems in lots of domains (finance, pharmaceutic, security, public health). However, those technologies are also the center of a polemic. Could those innovative techniques be a threat for privacy ? Indeed, without being aware of it, Data Mining can get into our private life by analyzing data related to our habits and thus extract from it facts we do not want that somebody knows! Thus, the use of such knowledge techniques should perhaps be modulated!

Bibliography

[AHKV98] Helena Ahonen, Oskari Heinonen, Mika Klemettinen, and A. Inkeri Verkamo. Applying data mining techniques for descriptive phrase extraction in digital document collections. In *Advances in Digital Libraries*, pages 2–11, 1998.

[BB99] Daniel Boley and Vivian Borst. Unsupervised updating of classification tree in a dynamic enviroment. In Oren Etzioni, Jörg P. Müller, and Jeffrey M. Bradshaw, editors, *Proceedings of the Third International Conference on Autonomous Agents (Agents'99)*, pages 390–391, Seattle, WA, USA, 1999. ACM Press.

[BM92] K. P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets, 1992.

[Bur98] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition, 1998.

[CL01] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001.

[CV95] C. Cortes and V. N. Vapnik. Support vector networks, 1995.

[Dix97] M. Dixon. An overview of document mining technology, 1997.

[DLTV05] Thierry Despeyroux, Yves Lechevallier, Brigitte Trousse, and Anne-Marie Vercoustre. Expériences de classification d'une collection de documents xml de structure homogène. In *Extraction et gestion des connaissances*. Cépaduès, 2005.

[FD95] Ronen Feldman and Ido Dagan. Knowledge discovery in textual databases (kdt). 1995.

[FPSM91] William J. Frawley, Gregory Piatetsky-Shapiro, and Christopher J. Matheus. Knowledge discovery in databases: An overview. In *Knowledge Discovery in Databases*, pages 1–30. AAAI/MIT Press, 1991.

[FPSS96] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery: An overview. In *Advances in Knowledge Discovery and Data Mining*, pages 1–34. 1996.

[GNF99] L. Goffinet and M. Noirhomme-Fraiture. Automatic cross-referencing of hci guidelines by statistical methods. In *Interacting with Computers 12*, pages 161–177. 1999.

[Gri97] Ralph Grishman. Information extraction: Techniques and challenges. In *SCIE*, 1997.

[KRMS01] Gunnar Ratsch Koji Tsuda Klaus-Robert Müller, Sebastian Mika and Bernhard Schölkopf. An introduction to kernel-based learning algorithms, 2001.

[KWD97] N. Kushmerick, D. Weld, and R. Doorenbos. Wrapper induction for information extraction. In *Intl. Joint Conference on Artificial Intelligence (IJCAI)*, pages 729–737, 1997.

[Mah03] Pierre Mahé. Noyaux pour graphes et support vector machines pour le criblage virtuel de molécules, 2003.

[MT96] Heikki Mannila and Hannu Toivonen. Discovering generalized episodes using minimal occurrences. In *Knowledge Discovery and Data Mining*, pages 146–151, 1996.

[Mus99] I. Muslea. Extraction patterns for information extraction tasks: A survey, 1999.

[NF] M. Noirhomme-Fraiture. Course of human computer interaction.

[PCC03] Antonio Badia Patricia Cerrito and James Cox. The application of text mining software to examine coded information, 2003.

[Ran97] Ralf Rantzau. Extended concepts for association rule discovery. Master's thesis, 11 1997.

[Ril93] E. Riloff. Automatically constructing a dictionary for information extraction tasks, 1993.

[Sod98] S. Soderland. Learning information extraction rules for semi-structured and free text, 1998.

[SS] Christos Stergiou and Dimitrios Siganos. Neural networks.

[Tan99] A. Tan. Text mining: The state of the art and the challenges, 1999.

[Vap95] V. N. Vapnik. The nature of statistical learning theory, 1995.

[Vap98] V. N. Vapnik. Statistical learning theory, 1998.

[Zaï99] Osmar R. Zaïane. Principles of knowledge discovery in databases. Technical report, University of Alberta, 1999.

[ZSD05] Debbie Zhang, Simeon J. Simoff, and John Debenham. Exchange rate modelling using news articles and economic data. 2005.

Appendixes

Appendix A

DropRedundantNews Method

```
private void dropRedundantNews(int NbrDays, int mappingNumber,
    int lengthDroppedWords){
  for (int b=0; b < this.singleDayNewsArray.length; b++){
    this.reciprocalDrop = new Vector();
    NewsItem[][] news = new NewsItem[NbrDays][];
    for(int c=0; c<NbrDays; c++){
      if(b+c < singleDayNewsArray.length){
        news[c] = newsDBManager.query("SELECT * FROM TrainingNews " +
            "WHERE Date = '"+singleDayNewsArray[b+c].getDate()+"'");
      }
    }
    redundantNewsMining(news,mappingNumber,lengthDroppedWords);
  }
}

private void redundantNewsMining(NewsItem[][] newsItems, int mappingPercentage,
    int lengthDroppedWords){
  Vector[][] vector = new Vector[newsItems.length][];
  tabTitles = new String[newsItems.length][];
  for(int i=0; i<newsItems.length; i++){
    vector[i]= new Vector[newsItems[i].length];
    tabTitles[i] = new String[newsItems[i].length];
    for(int j=0; j<newsItems[i].length; j++){
      tabTitles[i][j] = new String("");
      this.tabTitles[i][j] = newsItems[i][j].getHeadline();
      StringTokenizer st = new StringTokenizer(newsItems[i][j].getHeadline(), " ");
      vector[i][j]= new Vector();
      while (st.hasMoreTokens()){
        String elt = new String("");
        elt = st.nextToken();
        elt = elt.replace("'"," ");
        elt = elt.replace(","," ");
        elt = elt.replace("`"," ");
        elt = elt.replace("."," ");
        elt = elt.replace(";"," ");
        elt = elt.replace(":"," ");
        if(elt.length()>lengthDroppedWords){
          vector[i][j].add(elt);
        }
```

```
      }
    }
  }
  for(int a=0; a<vector.length; a++){
    for(int b=0; b<vector[a].length; b++){
      String[] tab = new String[vector[a][b].size()];
        vector[a][b].toArray(tab);
        for(int e=a;e<vector.length; e++){
          for(int f=0; f<vector[e].length; f++){
            String[] tab2 = new String[vector[e][f].size()];
            vector[e][f].toArray(tab2);
            if(!(this.tabTitles[a][b].equals(this.tabTitles[e][f]))){
              boolean drop =true;
              Iterator iter = this.reciprocalDrop.iterator();
              while(iter.hasNext()){
                String temp = new String("");
                temp = (String)iter.next();
                if(temp.equals(a+"-"+b+"-"+e+"-"+f)){
                  drop = false;
                }
                if(temp.equals(e+"-"+f+"-"+a+"-"+b)){
                  drop = false;
                }
              }
              if(drop){
                this.reciprocalDrop.add(new String(a+"-"+b+"-"+e+"-"+f));
                this.reciprocalDrop.add(new String(e+"-"+f+"-"+a+"-"+b));
                dropIfEnoughRedondance(tab,tab2,mappingPercentage,
                  this.tabTitles[e][f]);
              }
            }
          }
        }
    }
  }
}

private void dropIfEnoughRedondance(String[] t1, String[] t2,
    int nbrMapping, String title){
  int nbrSameWords = 0;
  for (int i=0; i<t1.length; i++){
          for (int j=0; j<t2.length; j++){
                      if (t1[i].equals(t2[j])){
                        nbrSameWords++;

                      }
            }
        }
        if (nbrSameWords >= nbrMapping){
        title = title.replace("'","''");
  String request = "DELETE FROM TrainingNews WHERE Title ='"+title+"'";
  this.newsDBManager.update(request);
  }
}
```

Appendix B

News Cutting (Big)

Fed	s	Bies	Says	Inflation	Under	Control	Rates
Will	Continue	to	Increase	Listen	Feb	Bloomberg	The
S	economy	is	strong	and	inflation	is	under
control	so	the	Federal	Reserve	will	continue	to
raise	rates	at	a	measured	pace	Governor	Susan
Bies	said	Inflation	as	measured	by	the	personal
consumption	expenditures	index	excluding	food	and	energy	is
between	percent	and	percent	a	year	Bies	said
On	average	prices	are	well	contained	she	said
in	a	speech	at	the	University	of	Tennessee
in	Martin	We	re	committed	to	a	measured
pace	of	continuing	to	raise	interest	rates	Bies
comments	mirrored	those	made	by	the	Fed	s
policymaking	Open	Market	Committee	after	last	week	s
meeting	The	FOMC	raised	the	benchmark	U	S
interest	rate	a	quarter	point	to	percent	and
said	it	would	continue	to	raise	rates	at
a	measured	pace	The	committee	also	said	inflation
is	well	contained	The	rise	in	oil	prices
in	the	last	year	got	the	Fed	s
attention	Bies	said	because	it	took	money	out
of	consumer	pocketbooks	that	would	have	been	spent
on	other	goods	and	boosted	the	U	S
trade	deficit	contributing	to	the	decline	in	the
dollar	Bies	said	that	with	the	large	trade
and	current	account	deficit	the	dollar	would	fall
further	if	it	were	not	being	boosted	by
foreign	governments	Our	trading	partners	intervene	aggressively	she
said	The	Chinese	are	pegging	their	currency	This
is	not	a	free	market	Bies	answered	questions
after	a	speech	at	University	of	Tennessee	in
Martin	She	said	the	U	S	s	till
attracts	a	tremendous	amount	of	foreign	direct	investment
that	helps	finance	the	trade	and	U	S
fiscal	deficit	The	federal	budget	deficit	which	helped
stimulate	the	economy	when	it	was	ailing	also
needs	to	be	brought	under	control	she	said
The	economy	now	is	growing	at	a	healthy
pace	and	you	don	t	need	that	extra
stimulus	she	said	Bies	said	President	George	W
Bush	s	proposal	to	divert	some	payroll	taxes
from	social	security	into	private	accounts	is	a
different	issue	from	making	the	retiree	program	solvent
She	said	most	people	are	not	educated	enough
to	responsibly	invest	in	private	accounts	You	can
t	just	offer	it	You	ve	got	to
really	dedicate	a	lot	to	consumer	education	she
said							

Table 6.4: News Cutting (Big)

StopWords (Big)

Fed	s	Bies	Says	Inflation	Under	Control	Rates
Continue	Increase	Listen	Feb	Bloomberg	economy	strong	inflation
control	Federal	Reserve	continue	raise	rates	measured	pace
Governor	Susan	Bies	said	Inflation	measured	personal	consumption
expenditures	index	excluding	food	energy	percent	percent	year
Bies	said	On	average	prices	said	speech	University
Tennessee	Martin	We	re	committed	measured	pace	continuing
raise	interest	rates	Bies	comments	mirrored	Fed	s
policymaking	Open	Market	Committee	week	s	meeting	FOMC
raised	benchmark	interest	rate	quarter	point	percent	said
continue	raise	rates	measured	pace	committee	said	inflation
well	contained	rise	oil	prices	year	got	Fed
s	attention	Bies	said	took	money	consumer	pocketbooks
spent	goods	boosted	trade	deficit	contributing	decline	dollar
Bies	said	large	trade	current	account	deficit	dollar
fall	boosted	foreign	governments	Our	trading	partners	intervene
aggressively	said	The	Chinese	pegging	currency	free	market
Bies	answered	questions	speech	University	Tennessee	Martin	said
attracts	a	tremendous	amount	foreign	direct	investment	helps
finance	trade	fiscal	deficit	federal	budget	deficit	helped
stimulate	economy	ailing	needs	brought	control	said	The
economy	growing	healthy	pace	don	t	need	extra
stimulus	said	Bies	said	President	George Bush	s	proposal
divert	payroll	taxes	social	security	private	accounts	different
issue	making	retiree	program	solvent	said	people	educated
responsibly	invest	private	accounts	You	just	offer	got
really	dedicate	lot	consumer	education	said		

Table 6.5: StopWords (Big)

Stemmer (Big)

fed	s	bi	sai	inflat	under	control	rate
continu	increas	listen	feb	bloomberg	economi	strong	inflat
control	feder	reserv	continu	rais	rate	measur	pace
governor	susan	bi	said	inflat	measur	person	consumpt
index	exclud	food	energi	percent	percent	year	bi
said	on	averag	price	said	speech	univers	tennesse
martin	we	re	commit	measur	pace	continu	rais
interest	rate	bi	comment	mirror	fed	s	policymak
open	market	committe	week	s	meet	fomc	rais
benchmark	interest	rate	quarter	point	percent	said	continu
rais	rate	measur	pace	committe	said	inflat	well
contain	rise	oil	price	year	got	fed	s
attent	bi	said	took	monei	consum	pocketbook	spent
good	boost	trade	deficit	contribut	declin	dollar	bi
said	larg	trade	current	account	deficit	dollar	fall
boost	foreign	govern	our	trade	partner	interven	aggress
said	the	chines	peg	currenc	free	market	bi
answer	question	speech	univers	tennesse	martin	said	attract
a	tremend	amount	foreign	direct	invest	help	financ
trade	fiscal	deficit	feder	budget	deficit	help	stimul
economi	ail	need	brought	control	said	the	economi
grow	healthi	pace	don	t	need	extra	stimulu
said	bi	said	presid	georg	bush	s	propos
divert	payrol	tax	social	secur	privat	account	differ
issu	make	retire	program	solvent	said	peopl	educ
respons	invest	privat	account	you	just	offer	got
realli	dedic	lot	consum	educ	said	expenditur	

Table 6.6: Stemmer (Big)

Appendix C

Effects on the Euro

This table presents the different factors concerning the Euro and their effects on the US dollar. The column "good/bad" indicates if it is classified as "good" or "bad" for the US dollar.

Factor	Factor effect	good/bad
interest rate goes up in Europe	Euro goes up	bad
interest rate goes down in Europe	Euro goes down	good
unemployment rate goes up in Europe	Euro goes down	good
unemployment rate goes down in Europe	Euro goes up	bad
gross national product (GNP) goes up in Europe	Euro goes up	bad
gross national product (GNP) goes down in Europe	Euro goes down	good
gross domestic product (GDP) goes up in Europe	Euro goes up	bad
gross domestic product (GDP) goes down in Europe	Euro goes down	good
inflation goes up in Europe	Euro goes down	good
inflation goes down in Europe	Euro goes up	bad

Table 6.7: Classification factors and their effects for the Euro